

# **Design and Implementation of Small Scale Joint Controllers for a Humanoid Robot**

*By*

***TIMOTHY JOHN CARTWRIGHT***

***Department of Information Technology and Electrical  
Engineering.***

***University of Queensland.***

Submitted for the degree of  
Bachelor of Engineering (Honours) in the division  
of Electrical Engineering.

19<sup>th</sup> of October 2001.

128 Trawalla St  
The Gap, QLD, 4061.  
19 October, 2001.

Professor Simon Kaplan,  
Head of School of Information Technology and Electrical Engineering,  
University of Queensland,  
St. Lucia, QLD, 4072

Dear Professor Kaplan,

In accordance with the rules of the degree of Bachelor of Engineering (Honours) in the division of Electrical Engineering, I submit the following thesis entitled “Design and Implementation of Small Scale Joint Controllers for a Humanoid Robot”. The work for this thesis was performed under the supervision of Dr Gordon Wyeth.

I declare that the work presented in this thesis is of my own doing and has not been previously submitted at the University of Queensland or any other institution. All material that is not my work has been referenced in the text.

Yours sincerely,

Timothy Cartwright

## **Abstract**

This thesis describes the design and development of the distributed motor control system for use in the GuRoo humanoid robot. Specifically this thesis deals with the upper body systems and communication protocols. The GuRoo robot has had no previous work carried out on it and as such, the design of all components was done in parallel. Limited humanoid projects are currently being undertaken worldwide and this is the first to be attempted at the University of Queensland.

The design described in this thesis covers the electrical design and testing of the RC servo motor controller board for the upper body control of GuRoo. Each controller board uses a Digital Signal Processor (DSP) for control of each system of joints. Previous work has involved each joint having a separate controller but because of space constrictions within a humanoid each board is designed to control multiply joints. The reasons behind the decisions for this configuration are given as well as full details of the solution obtained.

Within this thesis, the operating software for the joint control is detailed along with the communication to the iPAQ via the Controller Area Network (CAN) protocol. The servo control system has been tested and the responses under different conditions are discussed. CAN communications are still in testing with further work required to integrate the network of all the joints with the iPAQ.

## **Acknowledgements**

I wish to thank these following people for their help in completing this project.

Gordon Wyeth, for his supervision and ideas for the future of the project. His vision for what many of us thought as the impossible and his acceptance of things when they stuffed up.

Jarad Stirzaker, for being a good partner and sharing in the spoils of the project, but mainly because I guess I have to put him here.

James Kennedy, for his previous work in this area and support whenever it was needed.

David Finn, for the kind loan of his thesis project to help resolve technical difficulties with ours.

Everyone in the Robotics Lab for their “support and encouragement” during the length of this project, especially the humanoid team which has grown to be more than thesis group.

# Table of Contents

Abstract .....	iii
Acknowledgements .....	iv
List of Figures and Tables.....	vii
Chapter 1 – Introduction .....	7
1.1 Thesis Overview.....	2
1.2 The “GuRoo” Team .....	3
1.3 Scope of work for this Thesis.....	4
1.4 Chapter Outlines.....	4
Chapter 2 – Review of Current Technologies.....	6
2.1 Biped Humanoid Walkers .....	6
2.2 Distributed Control Systems .....	8
2.3 Controller Area Network (CAN) .....	9
2.4 DSP controllers.....	10
Chapter 3 – Device Specifications .....	11
3.1 Ideal product.....	11
3.2 System Specifications .....	11
Chapter 4 – Hardware Implementation.....	14
4.1 Board Overview .....	14
4.2 DSP Controller .....	15
4.3 Sensors & Support Circuitry .....	16
4.4 Programming Hardware .....	17
4.5 Network Interface.....	19
4.6 Power Supply & Motor Drivers .....	21
4.7 Board layout considerations.....	22
Chapter 5 – Software Implementation .....	24
5.1 Software Progress.....	24
5.2 Software Implementation.....	24
5.3 DSP Setup.....	25
5.4 Interrupt Service Routines.....	25
Chapter 6 – Evaluation and Future Developments .....	27
6.1 Project Performance .....	27
6.2 Future Work .....	28
Chapter 7 – Conclusions .....	29
References .....	30
Appendices.....	32

# Appendices

## Appendix A – Schematic Diagrams

A1 - Serial Programmer Schematic.....	A1
A2 - Joint Controller Schematic.....	A2

## Appendix B – PCB Layouts

B1 - Serial Programmer Layout.....	B1
B2 - Joint Controller Layout.....	B3

## Appendix C – Code Listings

C1 - Serial Test Code.....	C1
----------------------------	----

## Appendix D – Semiconductor Data Sheets

D1 - TMS320F243A DSP.....	D1
D2 - MAX811 Reset Regulator.....	D4
D3 - HC244A Buffer.....	D6

## Appendix E – Unpublished Paper G. Wyeth et al.

E3 - Design of an Autonomous Humanoid Robot.....	E1
--	----

## List of Figures and Tables

Figure 1.1 – CAD model of GuRoo Robot

Table 1.1 – 2001 GuRoo Humanoid Theses

Figure 2.1 - GuRoo CAD Model

Figure 2.2 – Honda’s Bipedal robots from the E0 to P3

Figure 2.3 – CAN Average Ping

Figure 3.1 Upper / Lower Board Split

Figure 4.1 - Controller Block Diagram

Figure 4.2 – Programming Board with 8 Wire Interface

Figure 4.3 – CAN Module

Figure 4.4 – CAN External Circuitry

Figure 4.5 – CAN Message Frame

Figure 4.6 – Digital Voltage Regulation

Figure 4.7 – Motor Power Cut-off

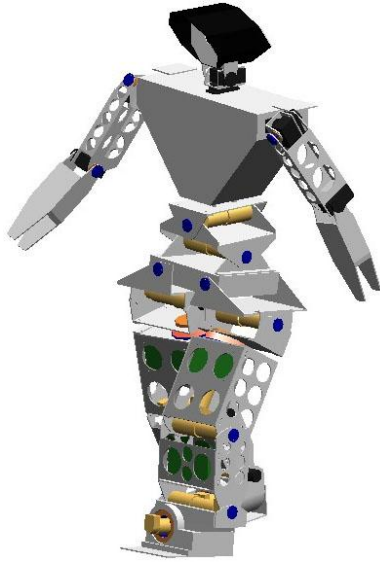
Figure 4.8 – Skeleton CAD model of upper body showing exact servo motor placement

Figure 4.9 – ‘Naked’ servo motor board showing layout of components

Figure 5.1 – Software flow chart

Figure 7.1 – GuRoo Upper Body Version 1

## 1.1 Thesis Overview



**Figure 1.1 - CAD model of GuRoo Robot**

The aim of this thesis is to describe the design and implementation of distributed small scale joint controllers for humanoid robot limbs. The robot in question “GuRoo” has been designed and built at the University of Queensland by a team of 12 undergraduate students. The design presented in this thesis consists of six joint controller boards, controlling a total of 23 joints, each connected through a serial bus to an iPAQ personal digital assistant (manufactured by Compaq).

Each joint controller unit comprises a Texas Instruments TMS320F243 Digital Signal Processor (DSP) which controls RC servo motors or operates an integrated half bridge to power DC motors. Each controller is shared between numerous joints and is located as close as possible to the areas it is controlling.

Specifically this thesis will detail the design of the upper body controller and the networking used to communicate between each controller and the iPAQ. Also described is the development of the DSP software for CAN communication. Jarad Stirzaker’s thesis [1] complements this one, and contains details of the lower body hardware and control loops. Work between these two projects was greatly collaborated enabling the designs completed to be easily integrated into the final humanoid.



## 1.2 The “GuRoo” Team

The work performed for the design of GuRoo (Grossly Under-funded Roo) has been undertaken by 12 undergraduate students with each student designing a different area of the humanoid (Table 1.1). With each member of the team concentrating on different aspects of the humanoid it would be advisable to browse all theses to get a full report on the project.

Student	Thesis
Shane Hosking	High Speed Peripheral Interface
Anthony Hunter	Mechanical Design of a Humanoid
Nathaniel Brewer	Power System for a Humanoid
Jarad Stirzaker	Design of DC Motor Controllers for a Humanoid Robot
Emanuel Zelniker	Joint Controllers for a Humanoid Robot Limbs
David Prasser	Vision Software for Humanoid Robot Soccer
Damien Kee	Design and Simulation of a Humanoid Drive System
Bartek Bebel	USB to CAN Bridge for Humanoid Project
Andrew Smith	Simulator Development and Gait Pattern Creation for a Humanoid Robot
Andrew Blower	Development of a Vision System for a Humanoid Robot
Mark Wagstaff	Mechanical Design and Internal Sensors for a Humanoid

**Table 1.1 – 2001 GuRoo Humanoid Theses**

The Robocup organisation summarises the objectives of many humanoid researchers with its goal, “By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champions.” The goal for the GuRoo project specifically has been to design and build a robotic bipedal humanoid capable of seeing an object e.g. soccer ball, walking over to the object, and kicking it. Progress thus far has not completed these goals however by the end of the project the team is confident of having a completed working design.

### **1.3 Scope of work for this Thesis**

The work completed for this thesis covers the electrical design, implementation and testing of the RC servo motor controller board for the upper body control of GuRoo. Incorporated into this is the networking software and protocol for communication between each controller and the central node. Work not completed for this thesis is the conversion from CAN to USB for communication with the iPAQ [2]. Work was also carried out on the lower body controllers (DC motor boards) in collaboration with Jarad.

### **1.4 Chapter Outlines**

*Chapter 2* details the previous work in the area of humanoid robotics and current trends in distributed control systems, humanoid bipeds, and controller networks. Also described is the history of the Controller Area Network (CAN) bus, its limitations and current developments. Brief background information on DSP controllers is also presented.

*Chapter 3* contains a description of the specifications for the electronics including the desired aims of the project. Preferred results for the project are also laid out giving full details to the limiting design decisions.

*Chapter 4* contains a description of electronic system through implementation to the final design. It describes the design of the upper body hardware including the DSP architecture, CAN hardware, and the driving circuitry for the actuators. Issues such as the physical dimensions and positioning of the hardware inside the robot are discussed, along with the problems of catering for a mechanical design not yet built. Also described are the reasons for designing for expandability and compatibility with all other modules of the robot.

*Chapter 5* describes the desired DSP software to be used to control each joint independently. Also giving a detailed explanation of the workings of the control system design including both the CAN bus and the communication protocol to be used. Problems with developing the software are mentioned and explained.

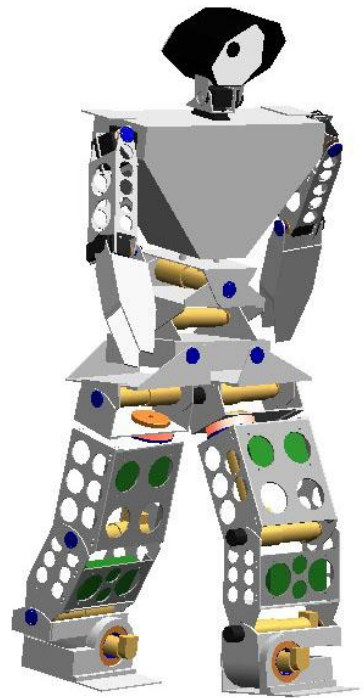
*Chapter 6* gives details of results obtained from the device. Included in this chapter are the results of both the software and hardware testing. Descriptions of the problems encountered during the design and possible future improvements are included in this chapter.

*Chapter 7* concludes on the system design and indicates how well the device performed in relation to the specifications in Chapter 3.

## Chapter 2 – Review of Current Technologies.

### 2.1 Biped Humanoid Walkers

The GuRoo, pictured in Figure 2.1, is a 24 degree of freedom bipedal walking robot currently under construction by the University of Queensland. With design and construction of the robot starting in March 2001, it is still in its infancy compared with many humanoid projects. The robot is 1.2 metres tall, 40 kilograms in weight and of distinct humanoid proportions. Actuators for the joints comprise 42V DC motors and 7.2V RC servo motors for the lower and upper body respectively. Attached to each lower body joint are optical encoders for position feedback, while the upper body is void of any such dynamic control mechanism. The lower body is split up into groups of 3 motors with a TMS320F243 DSP chip controlling each of the groups, while all 8 servo motors in the upper body are controlled by one TMS320F243. Network communication between the DSP controllers is over a Controller Area Network (CAN) bus connected to a central Compaq IPAQ. The robot is run off 42V supplied by four Nickel Cadmium batteries on board. Vision is achieved through a CCD camera with filtering being done on a SH4 board also attached to the central IPAQ.

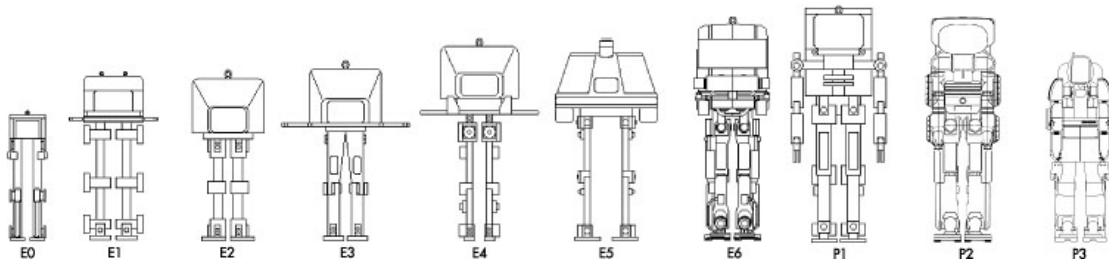


**Figure 2.1 - GuRoo CAD Model**

Research into bipedal robots can be separated into two main areas, dynamic and static walking, with the GuRoo falling into the first category. Research into dynamic bipedals has been going on since the 1970's and the first dynamic bipedal humanoid, WABOT-1 was created in 1973 by the Advanced Research Center for science and Engineering at Waseda University in Tokyo [3]. Increases in the number of research projects in humanoid robotics have led to many advances both in mechanical design and in the flexibility of walking algorithms. Honda's P3 and Sony's SDX are both good examples of current working humanoid robots. Honda's P3 can walk, turn, balance and navigate up and down inclines [4]. Sony's SDX-3 can achieve the same things as Honda's P3,

however it is noticeably smaller (20 inches tall) and has almost no technical information available on the design.

Honda's progression from bipedal robots to bipedal humanoid robots can be seen in Figure 2.2. Advances in research have allowed complex designs to be created, each one becoming more humanoid in shape. One of the problems with building full humanoid



**Figure 2.2 – Honda's Bipedal robots from the E0 to P3**

robots is that the space required to store all the actuators and electronics becomes larger than that available to keep the design within the boundaries of a humanoid shape. This problem of miniaturisation has led to designs like GuRoo where the shape has had to be modified, to fit all the hardware, and becomes bulky in comparison to the average humanoid figure [5].

Current research into humanoid robotics relies mostly on each joint in the system being controlled by separate controllers each connected to the central computational unit. This distributed control system better enables dynamic control algorithms to be run at high speed at the joints leaving the central node with free resources for the gait algorithms and other calculations.

## **2.2 Distributed Control Systems**

Distributed control systems for a robot offer several advantages over conventional central control. A robot can be constructed from standard actuators, each with its own microcontroller, connected to the main processing unit via a network. The only connections needed between actuators are the power and control signals which can be distributed along a serial bus. The main obstacle for design of such systems 20 years ago was the lack of small, high-speed microcontrollers, with low cost and power requirements, fast enough to run control loops and communicate over the network. With advances in microcontroller speed and a reduction in cost the first distributed control system was designed in 1984 by ROVAC for use in a fibre glass production facility [6]. The system used control boards at each node on a robotic crane to help position and spray on fibre to backing. Each node was connected to the central computer via a RS-442 bus running at 9600 Bits per second. The controllers received their desired position 20 times per second and also returned feedback on current position at the same sampling rate. Limited industry support for “distributed” systems saw designs created where all control nodes were kept within a cabinet and wiring run from there to the actuators effectively making the system only pseudo distributed. Such designs have mostly been abandoned now with most systems opting for placing the electronics as close to the joint being controlled as possible.

Recent developments in the area have shown great increases in both speed and precision when modifying old type cabinet systems to newer distributed systems. The PUMA 560 arm controller work described by Kennedy, 1999 [7] details the results of modifying the PUMA arm to be controlled by numerous boards connected over a CAN bus. Each joint had a separate controller enabled for the cabling and electronics involved to be significantly reduced in volume. The added power of the new controllers increased the speed and accuracy of the robot such that the electronics designed could outperform the mechanical limitations of the robot. Kennedy used 6 networked TMS320F241 DSPs, each being able to run a PID control loop, convert an A/D input, calculate the response and deal with current limiting and PWM generation all at a frequency of 20khz. The work in this thesis closely follows on from the work of Kennedy.

Recently distributed servo controllers have been demonstrated in humanoid robots with good results. Munsang describes a Centaur (lower body of a horse, upper body of a human) based robot design with a multi-layered controller structure [8]. The Centaur was designed with 3 separate levels of distributed control, the lowest level being TMS320F240 DSP based servo boards which communicated via SPI to the second level controllers. The second layer of control was TMS320C30 DSP chips which generated the trajectory for each of the lower controllers. The highest control node was the MVME162 CPU board connected via a VME bus to the second level of the system. Such a tiered control system design led to acceptable results whilst still enabling the controllers to be situated close to the joints being controlled. Numerous other servo motor control boards have been created, based around almost every controller available from the HC11 up to the SH3.

### ***2.3 Controller Area Network (CAN)***

The CAN bus is a high speed, low error communication standard developed for the automotive industry by Robert Bosch GmbH in 1982 [9]. It was developed because of difficulties in connecting or sharing data among ECUs (Electronic Control Units) in vehicles. Standard UART transmission is only suitable for point-to-point transmission, and multiple nodes are not allowed. One of the major benefits of CAN is that it is a multi-master communication system, allowing any node to broadcast messages that will not cause conflicts on the bus. The priority setting on messages allows for communication to be scheduled so that important information is always transmitted unhindered. Arbitration of the bus is done dynamically with no message being delayed whilst the new bus master is calculated. All lower priority messages are stopped until the high priority message has been transmitted and the bus is once again idle. The standard CAN protocol has an 11 bit identifier and hence can handle up to 2048 different nodes, with the extended CAN2.0B protocol a 29 bit identifier is available enabling over 500 million nodes to be accessed. Each CAN message can transmit from 0 to 8 bytes of data with segmentation used to transmit longer messages. The maximum reliable transmission rate is specified as 1 Mbit/s for networks with up to 40 meters between the two end nodes. Longer distances require that the data rate be reduced to

ensure there are not significant errors. Figure 2.3 shows the average response time of CAN bus “pings” with respect to the data rate specified for the network.

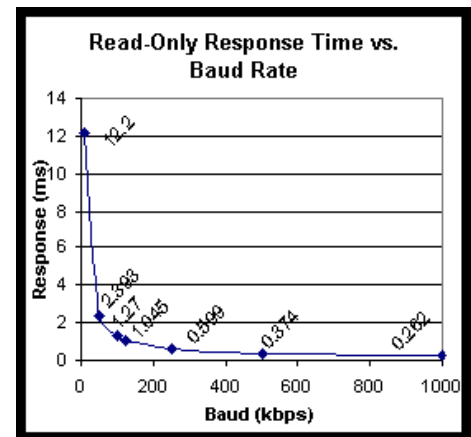


Figure 2.3 – CAN Average Ping

## 2.4 DSP controllers

The advances made in Digital Signal Processors make the devices very well suited for motor control applications. Many new DSPs are being advertised as being “designed for digital motor control” because of the large list of features associated with the devices. The power and flexibility of current DSPs allow for high speed control software to be run with ease. Current DSPs are highly integrated devices with respect to the large number of onboard peripherals now almost standard. Internal Flash memory, Analog to Digital converters, quadrature decoders, numerous PWM outputs, and integrated CAN modules make DSP chips a logical choice for GuRoo. Many controllers have incorporated peripherals, like those mentioned above, allowing for easy integration into both the DC motor and RC servo controller boards. High frequency PWM and encoder feedback are highly beneficial features available in a large amount of currently available DSPs. The high speeds are of little use when controlling RC servo motors which require a signal of between 45-50 Hz. The choice of DSP had to reflect both these extremes in operation. Controlling RC Servo motors makes little use of the powerful DSP resources available, however control of DC motors with the associated control loops requires such processing power.



## Chapter 3 – Device Specifications

### 3.1 Ideal product

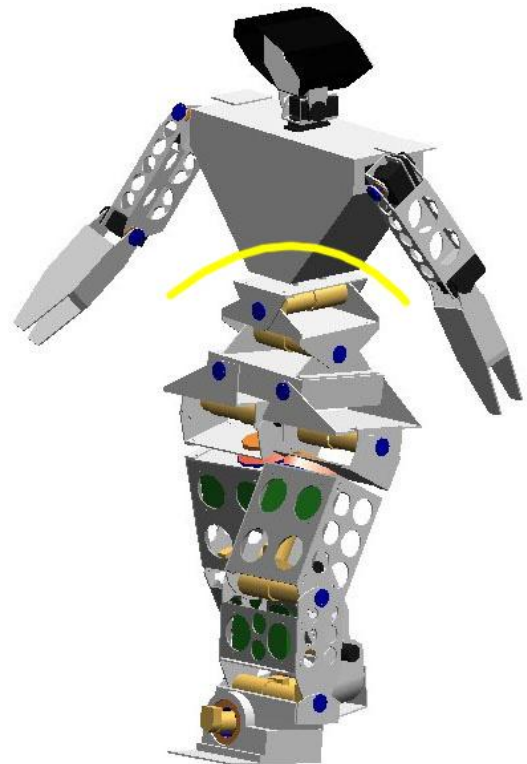
The GuRoo, being an entirely new project for the university, has very few similar products to be compared with. International universities and companies do have working prototypes of humanoids, however most of the details behind the internal workings are not publicly available. The ideal product is therefore very difficult to define as there are many research areas being pursued and the ‘perfect’ humanoid joint control system is only limited by funds and design time. There are however some broad aspects which can be defined.

Ideal humanoid joint control systems should fulfil these criteria:

- Distributed boards close to joints limiting wiring.
- High speed control loops should be possible.
- Low power consumption.
- Minimal physical size.
- High speed, low error networking with limited cabling between controllers.

### 3.2 System Specifications

The mechanical design and choice of motors for each of the joints greatly affected the specifications for the controller boards. The upper and lower body boards are defined by the use of RC servo motors for the lightweight upper joints and 40V DC motors for the lower joints respectively. Using the ideal product defined above as a baseline the specifications for the GuRoo’s joint controller system are split into two separate categories. The upper body and lower body required vastly different motor control systems, however to help reduce costs a lot of overlapping was done between the two distinct modules.



**Figure 3.1 Upper / Lower Board Split**

Specifications identical between both boards were:

- Need for high speed control loops.
- Low power consumption.
- Minimal physical size.
- High speed, low error networking with limited cabling between controllers.
- Expandability.

The differences between the upper and lower body boards are related only to the motor driver and feedback system. The lower body boards as mentioned previously are described in detail in Jarad's thesis [1] however; a brief overview of their function is required here. The lower body joints require fast controllers to be able to handle the high-speed control loops. The controllers must also have numerous integrated peripherals to deal with the current and encoder feedback from the DC motors. Quadrature decoding capability and fast current sensing via analog to digital capture are both requirements specific to the lower joints. These requirements were duplicated for the upper body to help alleviate the problem of working with two different controllers. The upper body by itself requires no special capabilities aside from those already described for the overall solution and the lower body.

To be able to meet these requirements the controller has to be both of high speed and be able to deal with the complexities of control loops involving feedback. Because of such requirements, 8-bit processors were unacceptable leaving 16 or 32 bit available. High-end devices such as SH3 were excluded from the design because of the requirements of a low cost solution.

Size requirements restricted the boards to being only as large as the spaces in the robot could handle. These spaces were changed numerous times during construction and hence a criterion was decided upon for the size of the boards. The lower body boards were to be approximately the same size as the boards produced by Kennedy [7] during his PUMA 560 thesis. The upper body board had plenty of space to 'fit', however an optimistic 3" by 3" size was decided. The boards not only had to be small enough to fit inside the robot but also cabling had to be minimised to ensure everything could be interconnected with little problems.

With the requirement of small amounts of cabling, each board should be positioned as close as possible to the joint it was controlling. High-speed joint control requires feedback to the central host to be continuous and without error. The network for the communication between nodes therefore had to have a high bandwidth, low error rate and minimal cabling associated with it.

## Chapter 4 – Hardware Implementation

### 4.1 Board Overview

The specifications defined in the previous chapter led to a board design that can be split into 5 distinct sections as shown in figure 4.1 below. Each segment is described in detail in the rest of this chapter.

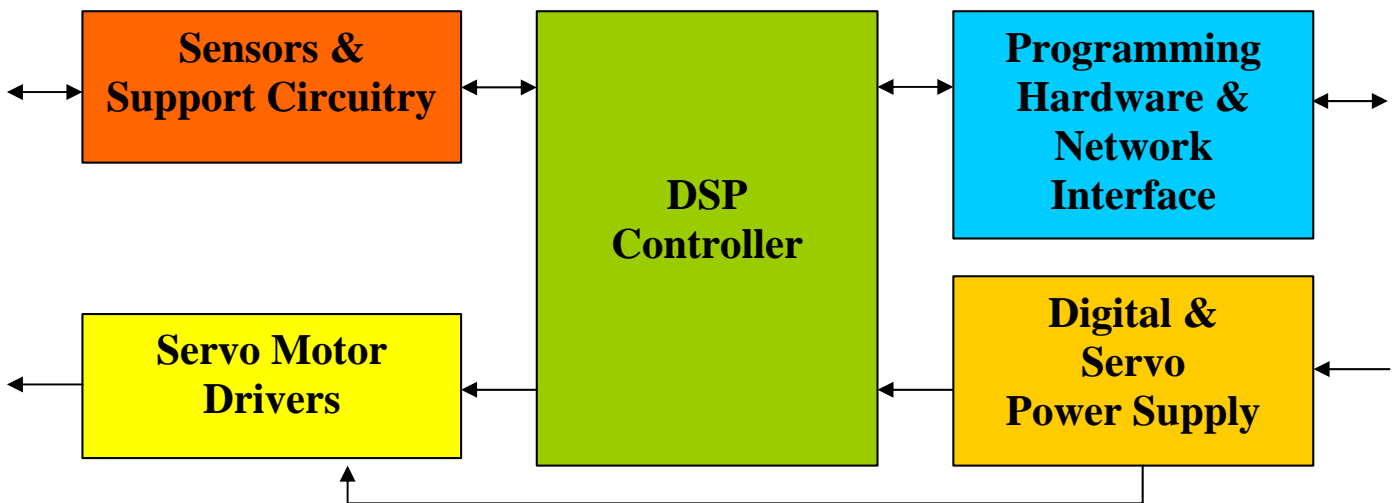


Figure 4.1 - Controller Block Diagram

Because of physical size restrictions in the arms of the robot the design was forced to be based around one central controller that controls all upper body servo motors. This contradicts the specification of having the boards as close to the joints as possible however it further reduces the wiring and so is acceptable. The one board controls all 8 RC servos from a central position in the chest cavity. More details of the size and layout constraints are mentioned later in the chapter.

## **4.2 DSP Controller**

The requirements specific to the upper body board leave many choices available for the controller. Factoring in the needs of the lower body power boards limits the choice dramatically. The requirement for high-speed control loops with encoder feedback lead the design towards chips which are specifically designed for these applications e.g. DSP's from Texas Instruments or Analog devices. Another chip that had the processing power required was the MC68376. The MC68376 was a favourable choice with its Time Processing Unit (TPU) allowing for fast quadrature decodes and PWM output capabilities. The major drawback to the '376 was the lack of internal flash memory which would have increased the space required on board and the lead time in purchasing of the devices.

Higher speed devices as mentioned previously would require too much external circuitry and lack the required peripherals for the task. The Hitachi SHx series of devices could easily perform the tasks required, however their high cost and excess peripherals made them an unsuitable choice. The Texas Instruments TMS320F24x series of devices was the most suitable out of the range of DSP's available [10]. With the 24x's onboard 8k flash, serial programming capability and integrated CAN module being the some of the major advantages of using the device. The drawback to using the 24x devices is the lack of adequate quadrature decoding capability. Although not an issue with the upper body, which has no feedback, the lower body needed the ability to decode 3 separate encoder counts at once. The external circuitry required to be added to the lower body boards to complete the task effectively made the processor a more expensive choice than the MC68376 devices.

Within the 24x family of devices there were two choices available, the TMS320F241 or the TMS320F243. The significant difference between the two devices was the external memory addressing capability of the 'F243. This was not required by the upper body board but for the external quadrature decoding capabilities of the lower body board.

The final decision for the controller was the TMS320F243 DSP from Texas Instruments. The DSP is not ideally suited for the job, however features such as the 8

PMW outputs, CAN module, 8 channel Analog to Digital conversion and 20 MHz clock frequency made it the most attractive solution available within the time constraints. The reasons for choosing the TMS320F243 controller were primarily based around the lower body system as the upper body's requirements were met successfully by a broad range of devices.

### ***4.3 Sensors & Support Circuitry***

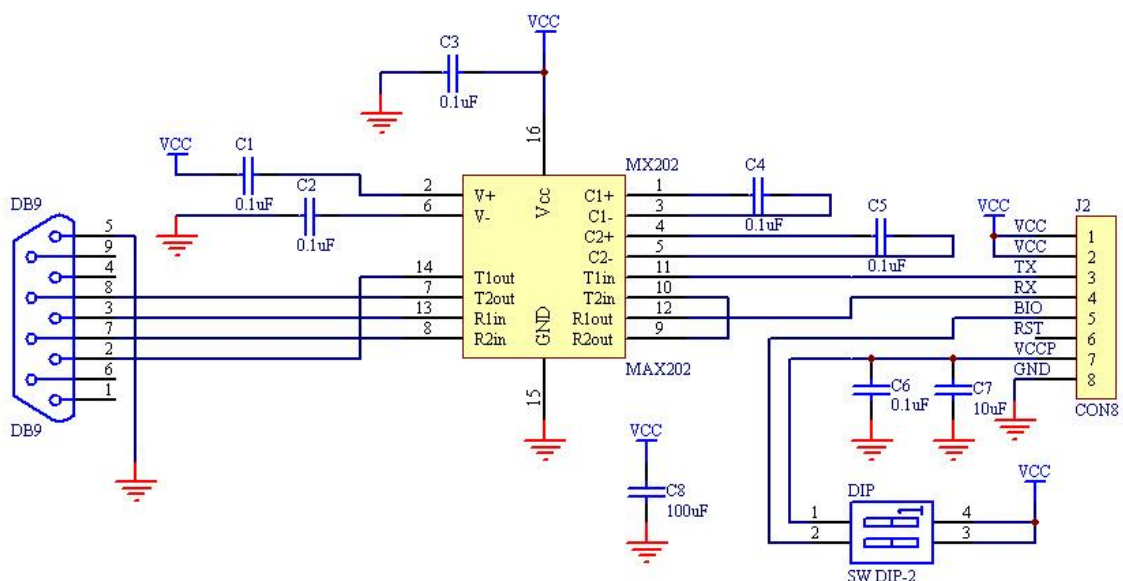
The original upper body was to be split between two different controller boards each with three accelerometers attached for position feedback of the body. The conglomeration of the two boards into one resulted in a board which needed to be able to support as much external circuitry as possible. During the design of the controller boards numerous features required from the boards changed. As such the boards were designed with maximum expandability in mind allowing for any of the features of the boards to be exploited. This expandability was achieved by making available on the board all 8 channels of the A/D converter through connectors along one of the edges. Along with the A/D converters part of IO port D was made available for visual testing of the device through 3 LED's. Currently the board does not require any of the external sensor inputs, however future uses might include the original accelerometers, temperature sensors or potentiometer feedback from modified Servo motors.

The DSP requires very limited support circuitry consisting of a 5 MHz crystal which is internally converted via the Phase Lock Loop (PLL) circuitry and a reset generation circuit. For the purposes of this project high processor speed was required so the four times multiplication of the PLL was exploited to gain a 20 MHz internal clock. Reset generation is achieved with the MAX811 IC from Maxim Semiconductors. The IC asserts a reset condition on the DSP reset pin whenever the VCC to the system drops below 93% of the usual 5V. This reset condition is asserted for a minimum of 140ms giving the DSP enough time to boot up safely without corrupting any of the internal memory or registers.

## 4.4 Programming Hardware

There are two main methods of programming the DSP's, the first method via serial connection to an internal bootloader and the second method by an external JTAG device. Ideally all programming would be done serially via the bootloader however having the DSP's as surface mount meant that any chip that malfunctioned could not simply be removed to be fixed in another board. The JTAG interface allows an external device to reprogram the DSP through dedicated hardware not associated with the bootloader. Reprogramming via the JTAG interface allows a new bootloader to be loaded into the Flash memory of the device there by permitting programming through the serial link again.

Serial programming with the bootloader requires the modification of two pins, namely the VCCP and BIO pins. When both the BIO and VCCP pins are high the device is placed into programming mode and awaits communication over its serial linkup. To save space on the main controller board the RS232 level conversion IC and switches for the BIO/VCCP pins were placed on a separate programming device. The connection between the programming board and the controller board is made through a small 8 wire interface as opposed to the large DB9 interface from the programming board to the PC. Figure 4.2 (below) shows the schematics for the programming board with the 8 wire interface to the controller board.



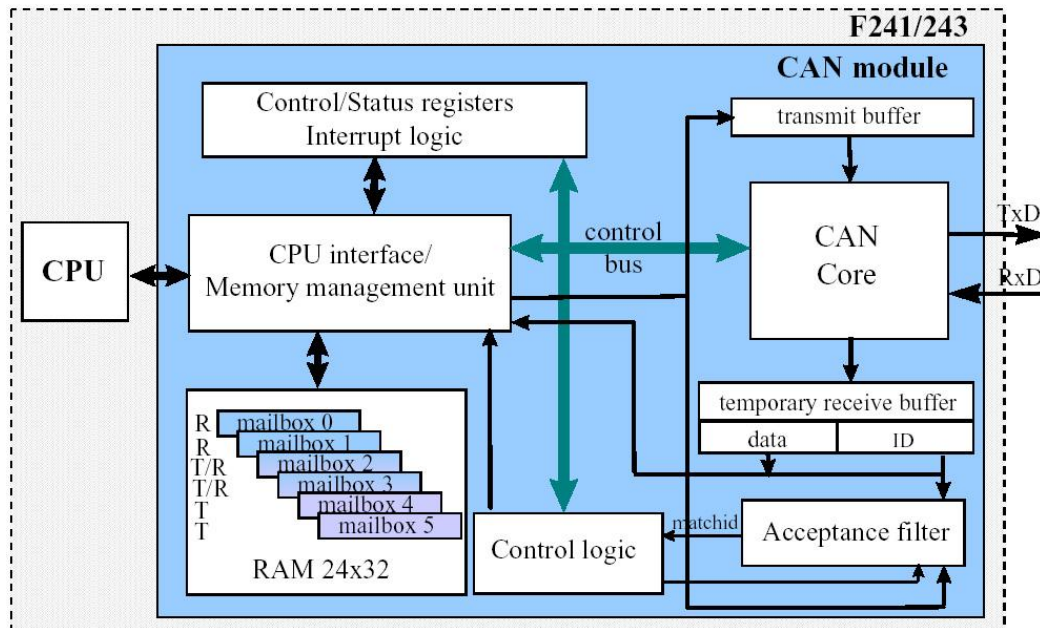
**Figure 4.2 – Programming Board with 8 Wire Interface**

To ensure that each board could be reprogrammed if an error occurred within the bootloader all 6 boards contain JTAG circuitry so they can interface with the Ice\*Pack JTAG device from Softronics [11]. The JTAG interface is a 13 wire interface connected directly to the DSP with limited external circuitry needed for operation (two pull up resistors on wires EMU0 and EMU1).



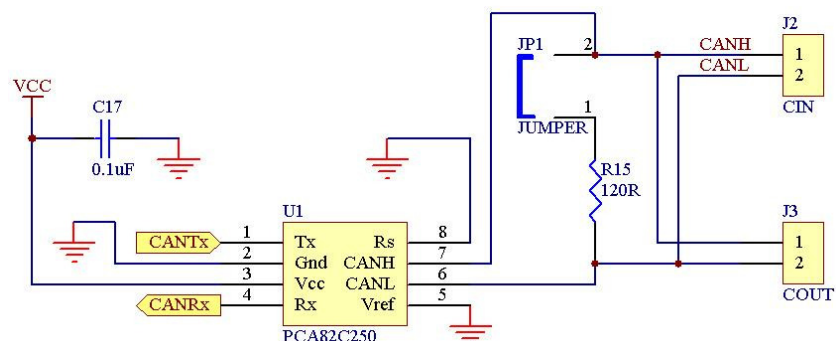
## 4.5 Network Interface

The 'F243 DSP contains a full CAN2.0B compliant module that handles all the low level networking in a separate CPU to the main core. This CAN module controls the 2 wire bus interface with all associated transmissions, error detection, arbitration, message filtering and acknowledgments (Figure 4.3).



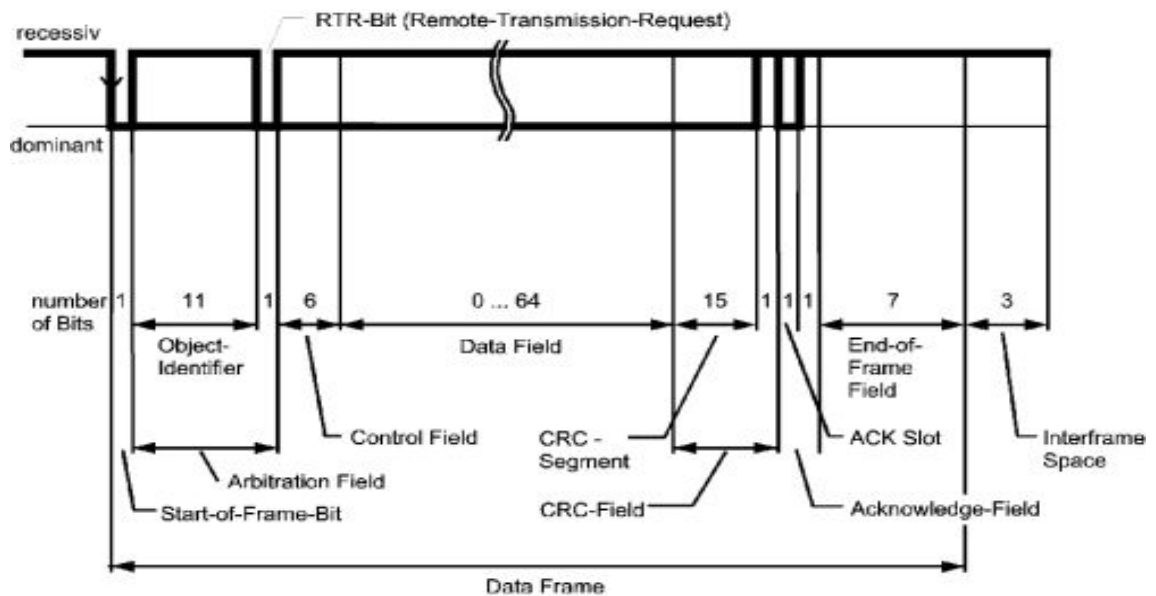
**Figure 4.3 – CAN Module**

Using only 2 wires the CAN system was well suited to the project limiting the wiring between each module. Using a multi-master system like CAN also allowed any node to be added onto the network with out having to run extra cabling from a central hub like a host-master network would have required. The bus wires CAN H and CAN L are effectively terminated at the end of the network by placing a 120Ω resistor interconnecting the two wires. Each board contains a jumper which can be connected to allow it to act as the network terminator (Figure 4.4).



**Figure 4.4 – CAN External Circuitry**

External to the DSP the CAN bus requires only the terminating resistor as mentioned and a level driver IC. During the design of the network the communication protocol devised would have required 1 message to be transmitted to each board 1000 times every second. Each message conforms to the standard CAN formatting (Figure 4.5) requiring at maximum 108 bits to be transmitted.



**Figure 4.5 – CAN Message Frame**

The transmission rate of the CAN bus required was then calculated.

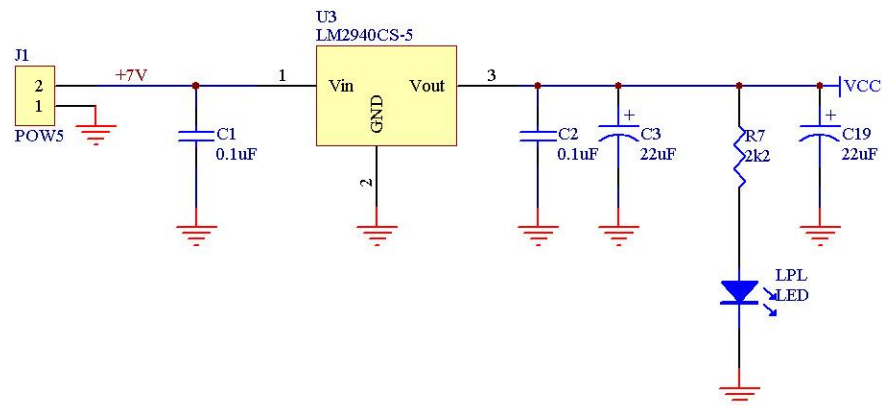
$$BaudRate = 7 \times 1000 \times 108$$

$$BaudRate = 0.756 \text{ Mbits / Sec}$$

Specification for the CAN bus allows transmission rates up to 1Mbit/Sec if the length of the bus is shorter than 40 meters and rest of the circuitry is able to handle that data rate [12]. Supplies of 1Mbit CAN driver chips could not be found within the time of the project so 0.5Mbit drivers were used. The use of slower speed driver chips limited the number of nodes available on the network however this has not hindered progress with the full humanoid not likely to be completed before demonstration day. Figure 4.5 shows the CAN external circuitry.

## 4.6 Power Supply & Motor Drivers

The joint controller board is powered by two separate supplies both of which run off 7.2v NiCad batteries. The low current logic power supply is distributed through the same eight core twisted wire as the CAN bus. This enables the further limiting of bulky power wires to each board. The problem of noise on the power line disturbing the CAN bus is minimal as the power consumption of the logic circuits is relatively constant compared with the switching power of the motor circuit. The logic supply is an unregulated 7.2v, coming from the main power supply board in the chest, and is regulated onboard using a low dropout 7805 equivalent regulator (Figure 4.6).

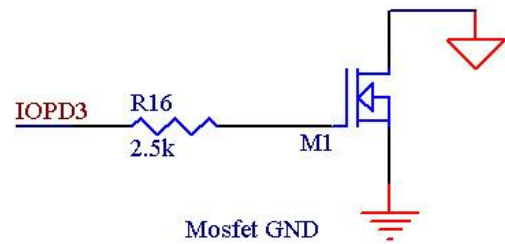


**Figure 4.6 – Digital Voltage Regulation**

Power to the RC servo motors is again from a 7.2v NiCad battery however this battery is separate from the digital supply. Connected via its own power line, this supply is unregulated at both ends of the system (power board and controller board). Using an unregulated supply to power the servo motors brings in extra considerations in the software with the actual voltage that the motors are running at being unknown.

If the logic power is lost from the servo controller board with the servo motor power is still on, a serious problem could occur. Each motor, which is no longer being provided a PWM signal to drive it to the correct position, would attempt to assume the default or 0% PWM position. This rapid change in direction and movements of the upper body would be very undesirable for safety reasons alone and to counteract such an event ever happening a logic shutoff was designed into the servo power supply. Each servo motor is connected to the power supply through a MOSFET which is driven by a spare IO pin

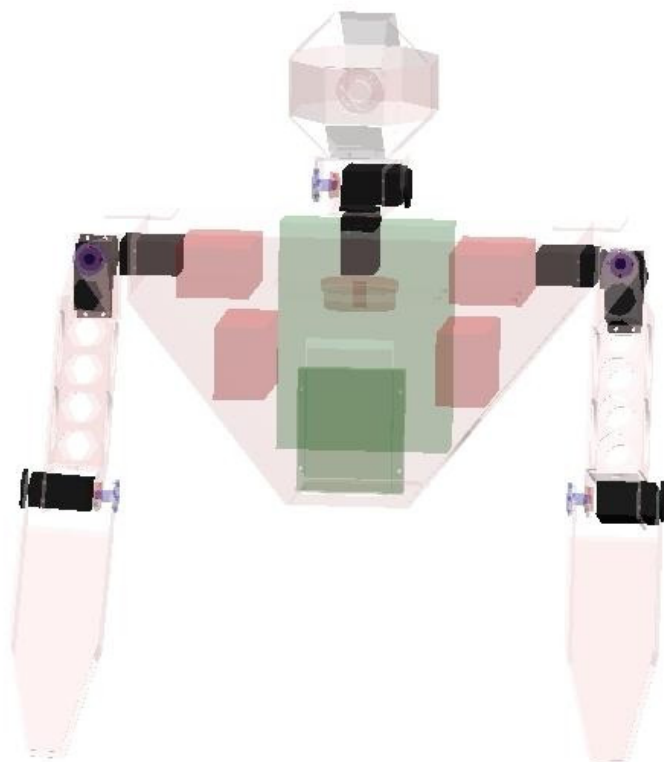
on the DSP (Figure 4.7). When a power error occurs the DSP has full control over the servo motors enabling them to be shut down by choice e.g. priority shutdown message over the CAN bus or when no power is supplied to the DSP the motors cannot return to the 0% position. The DSP can only source 8 milliamps which not enough current to drive the servo motors directly from the output pins so a simple buffer is placed in between each PWM output and the corresponding servo motor.



**Figure 4.7 – Motor Power Cut-off**

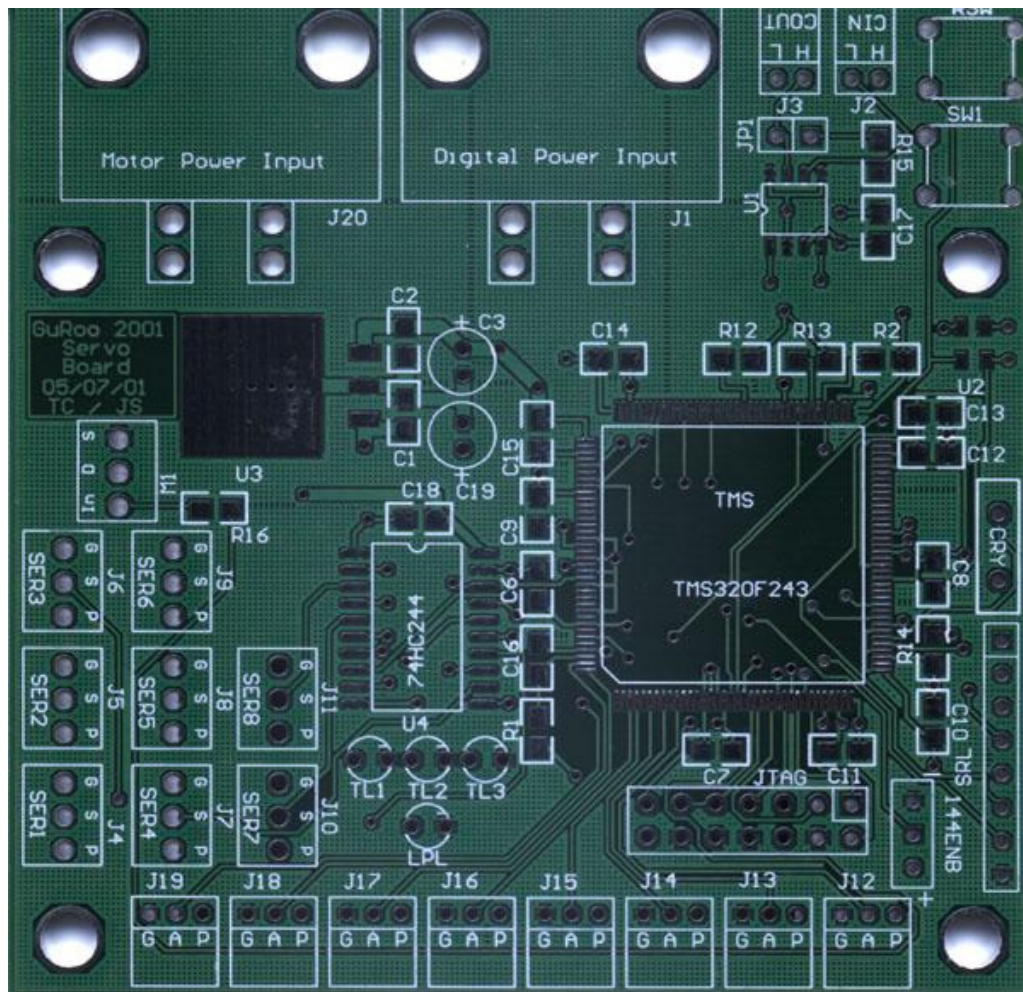
#### **4.7 Board layout considerations**

Size requirements restricted the boards to being only as large as the spaces in the robot could handle. These spaces were changed numerous times during construction and the device reflects these constant changes by being as compact as possible both in PCB area and in depth. Specifications for a 3” by 3” board proved feasible with the final design fitting in a 3” by 3” by 1” box. The placement of the board in the chest cavity enables it to be close to the servo motors it is driving, (Figure 4.8) however, it also makes it close to Nathaniel’s power supply board [13]. Placing the board so close to a high current and high frequency switching device requires that the system be well shielded. Top and bottom layers of the four-layer board contain large ground plains to help reduce any interference from external sources whilst the internal two layers are used to shield the DSP circuitry from the



**Figure 4.8 – Skeleton CAD model of upper body showing exact servo motor placement.**

switching of the servo motors. The reset circuitry and clock are also kept as far away from the power side of the board as possible to help reduce any brownouts or skewing of the clock signal from motor interference. To simplify wiring of the board into the robot all connections that must go to the central hub are situated along the bottom edge and all sensor connectors are mounted on the opposite edge. A brief illustration of the PCB layout is shown in (Figure 4.9).



**Figure 4.9 – ‘Naked’ servo motor board showing layout of components.**

- J19 - J12      Analog to digital connectors.
- J10 - J4        Serial motor connectors.
- J1 & J20        Digital and Motor power supply respectively.
- J2 & J3         CAN input and CAN output connectors.

# Chapter 5 – Software Implementation

## 5.1 Software Progress

Software progress was severely delayed during the project due to an error on our behalf. The DSP boards were deemed to be inoperable for a two month period whilst Jarad and I attempted to isolate the problem. Eventually the problem was traced to corrupted boot loaders in each of the DSP chips. Reprogramming the devices with a new bootloader through the JTAG interface fixed each board. This delay in the middle of the project has resulted in little of the proposed software being completed. This chapter defines the intended implementation that is currently being developed in time for demonstration day.

## 5.2 Software Implementation

There is no requirement for high speed motor switching and complex control loops in the upper body as each of the servo motors is switched at a frequency of 45-50Hz. The lack of feedback on the motors also limits the complexity of the resulting DSP code. When powered up, the device initiates a standard setup routine configuring all of the peripherals to the appropriate settings. Once completed the controller lies dormant in a null loop whilst awaiting an interrupt to trigger further code. Figure 5.1 (below) shows the basic flow diagram of the software system.

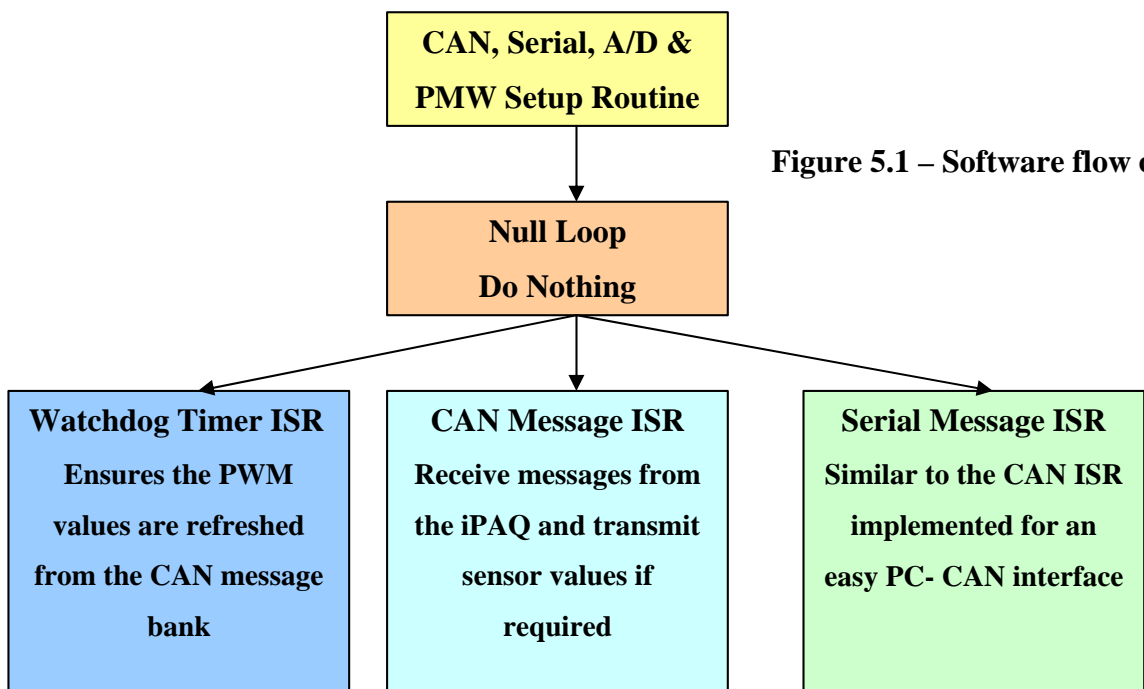


Figure 5.1 – Software flow chart

### **5.3 DSP Setup**

Prior to the DSP going into the null loop, certain peripherals must be enabled and set up for correct operation. Initially the interrupts are disabled until the setup is complete. The CAN module is configured to use mailboxes 2 and 3 to transmit and receive messages respectively, as well as setting up the interrupt on messages being received. Because the full humanoid is not yet completed, the data rate does not need to be the full 1Mbit/sec (calculations in Chapter 4) so the CAN is set at a lower bit rate to ensure no errors are encountered. Serial communications are set at a 38400 baud rate to enable PC debugging and control of the device. Similar to CAN, the serial connection is configured to interrupt on receiving a message. The PMW modules are configured for a 50 Hz output signal and the A/D converters are enabled to allow for sensor data to be collected if requested. Aside from all the peripherals the last 3 pins of port D are configured as outputs enabling the device to display its current state to the LEDs on the board. Interrupts are now enabled and the program enters into an infinite loop.

### **5.4 Interrupt Service Routines**

The receiving of an interrupt places the DSP in the interrupt service routine where it checks all the relevant flags to determine which interrupt has been triggered. When a communication interrupt arrives the software parses the message and enters a further routine based on what the message contained. A CAN communication, for example, can contain one of a few specific messages.

0. Emergency shutdown un-graceful.
1. Set the PMW value for motor X.
2. Read sensor value X and transmit.
3. Graceful shutdown.

Only message 2 requires transmission over the CAN bus back to the iPAQ for calculations. The controller board has no other transmit capabilities as there is no other feedback connected to the device.

Serial message packets currently involve the transmission of a single letter or number to the DSP to alter its state. After parsing, the message is checked for one of the known input conditions and the appropriate function loads to complete the task. Filtering of serial transmissions has to be done in software unlike the hardware identifier filter of the CAN bus. The resultant frequent interruptions are the reason for imposing the relatively slow serial baud rate compared with the 500Kbits/sec of the CAN link. Serial communication is for demonstration and debugging purposes only and is not a feature that will be incorporated into the final device.

Watchdog interrupts are used, as opposed to timer 1 overflows, in the design of the PWM software because of the limitations of the DSP in handling 8 separate PWM outputs. The 'F243 is designed primarily for driving half or full H-bridges and as such the 6 main PWM outputs are split into 3 groups of inverting and non-inverting pairs. Each pair has one pure PWM signal and one which is the inverse. This is of limited use in driving servo motors so alternative means for the driving had to be devised. The use of PWM outputs 1, 3 and 5 along with the independent PWM outputs 7 and 8 allows for 5 simultaneous signals to be generated. The need for 8 signals was resolved by using 3 pins from port D. The need for only a 50Hz PWM signal means that the cycle length of each pulse is 400,000 instructions. Allowing the watchdog to overflow every 2000 instructions leaves the device with an effective 200 possible PWM outputs at the required 50Hz output frequency. This use of the watchdog timer is preferred because it enables the use of Digital I/O lines to drive the servos.



## **Chapter 6 – Evaluation and Future Developments**

### ***6.1 Project Performance***

Programming hardware and the Servo Motor board work and can easily meet all the requirements set out for them. Hardware performance of the device is exceptional considering that there was no prototyping phase available during the beginning of the project due to the original 1<sup>st</sup> of August deadline. All peripherals on the board work as expected however limited numeric performance measures can be carried out on the system in its present state. The development of a board which can easily fit inside the humanoid with the ability to control the 8 servo motors simultaneously can be seen as a success.

The DSP and all its associated peripherals, A/D conversion, Serial in/out, Interrupt driven communication, have been tested. Full code to be able to test the final workings of the device has not yet been completely written. Serial communications to and from the board work allowing the demonstration of communication similar to that expected from CAN without the CAN working at the moment. The ability for the device to perform as required is evident. More software must be composed before a final demonstration is working. The lack of a working USB to CAN interface hindered progress in this area however workarounds with the serial linkup should enable the software to be completed before demonstration day.

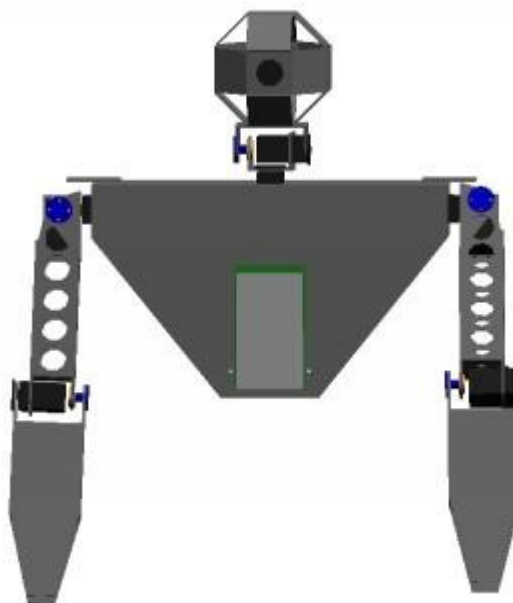
## 6.2 Future Work

- Much work needs to be completed on the software for the joint controller before it will be ready for use in the humanoid. For demonstration day however there will be a working device able to show the communication and motor driving code working fully.
- The entire network interface requires 3 separate theses to be completed; Bartek Bebel with the USB to CAN interface [2], Andrew Smith with the iPAQ network software and protocol [14] and myself with the controller end software.
- Revision of the choice of controller needs to be done to determine if a more suitable device can be found to do the job. The MC68376 would be a good device to start the review with.
- Provision for more in depth onboard testing or the production of a prototype board for the TMS320F243 would be most beneficial if future projects were to use the same controller.
- A PC interface to the GuRoo network needs to be created enabling easy debugging of the whole robot and each node independently. Ideally this could also be on the iPAQ delivering real time network status reports to the screen.
- Servo motor control is making very limited use of the DSP's power, perhaps some more challenging tasks such as adding numerous sensors and even potentiometer feedback to the device would give a better dynamic model of the humanoid.

## Chapter 7 – Conclusions

The aim of this thesis and the 11 other theses associated with the humanoid “GuRoo” was to construct a walking humanoid robot within a set budget and in under 1 year. Achieving this goal was a very ambitious one but the apparent failure to complete this should not be looked upon harshly. A full humanoid walking is not completed but the subsystems are in place for future work in the area to complete the original task. This specific thesis involved the creation of a motor controller board capable of driving the upper body which consists of 8 servo motors (Figure 7.1). The networking software for all the joint controller boards was also undertaken as an aim of this thesis.

This thesis was not entirely successful in achieving these goals with large setbacks forcing the software component to be left unfinished. The hardware for the controller works within the specifications and has been shown to be able to power a servo motor to a desired pre-defined position. Full integration of the hardware into the CAN network has not yet been achieved but the process for completing this has been outlined in Chapter 5. The task of building an entire humanoid is not a small one and can easily be the subject of future research within the department. Each component of the robot can be optimised if additional work is completed on the GuRoo project. Future work on the GuRoo should be encouraged allowing the research foundations of the first University of Queensland humanoid team to built upon.



**Figure 7.1 – GuRoo Upper Body Version 1.**

## References

- [1] J. Stirzaker, *Design of DC Motor Controllers for a Humanoid Robot*, University of Queensland Undergraduate Thesis, 2001.
- [2] B. Bebel, *Development of a USB to CAN Interface*, University of Queensland Undergraduate Thesis, 2001.
- [3] I. Kato et al, *Information-power machine with senses and limbs (Wabot 1)*, Symposium on Theory and Practice of Robots and Manipulators, 1973.
- [4] K. Hirai et al, *The Development of Honda Humanoid Robot*, IEEE Conference on Robotics and Automation, 1998.
- [5] G. Wyeth et al, *Design of an Autonomous Humanoid Robot*, Submission for the Australian Conference on Robotics & Automation Sydney, 14 - 15 November 2001.
- [6] *Arrangement comprising a system providing movement, processing and/or production*, PCT International Publication Number WO 85/01007, 1985.
- [7] J. Kennedy, *Design and implementation of a distributed digital control system in an industrial robot*, University of Queensland Undergraduate Thesis, 1999.
- [8] K. Munsang et al, *Development of a humanoid robot CENTAUR*, 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999.
- [9] ISO 11898: "*Road vehicles - Interchange of digital information - Controller area network (CAN) for high-speed communication*", 1993.
- [10] Texas Instruments, "*TMS320F243 Product Folder*,"  
<http://focus.ti.com/docs/prod/productfolder.jhtml?genericPartNumber=TMS320F243>  
(current Oct. 18, 2001).

[11] “*Ice\*Pack*,” <http://www.softronx.com/Products/Emulators/IcePack/icepack.htm>  
(current Oct. 18, 2001).

[12] Robert Bosch GmbH, “*CAN Specification 2.0*,” Postfach 50 D-7000 Stuttgart 1,  
1991.

[13] N. Brewer, *Power System for a Humanoid*, University of Queensland  
Undergraduate Thesis, 2001.

[14] A. Smith, *Simulator Development and Gait Pattern Creation for a Humanoid  
Robot*, University of Queensland Undergraduate Thesis, 2001.

# **Appendices**

## **Appendix A – Schematic Diagrams**

**A1 - Serial Programmer Schematic**

**A2 - Joint Controller Schematic**

## **Appendix B – PCB Layouts**

**B1 - Serial Programmer Layout**

**B2 - Joint Controller Layout**

## **Appendix C – Code Listings**

**C1 - Serial Test Code**

## **Appendix D – Semiconductor Data Sheets**

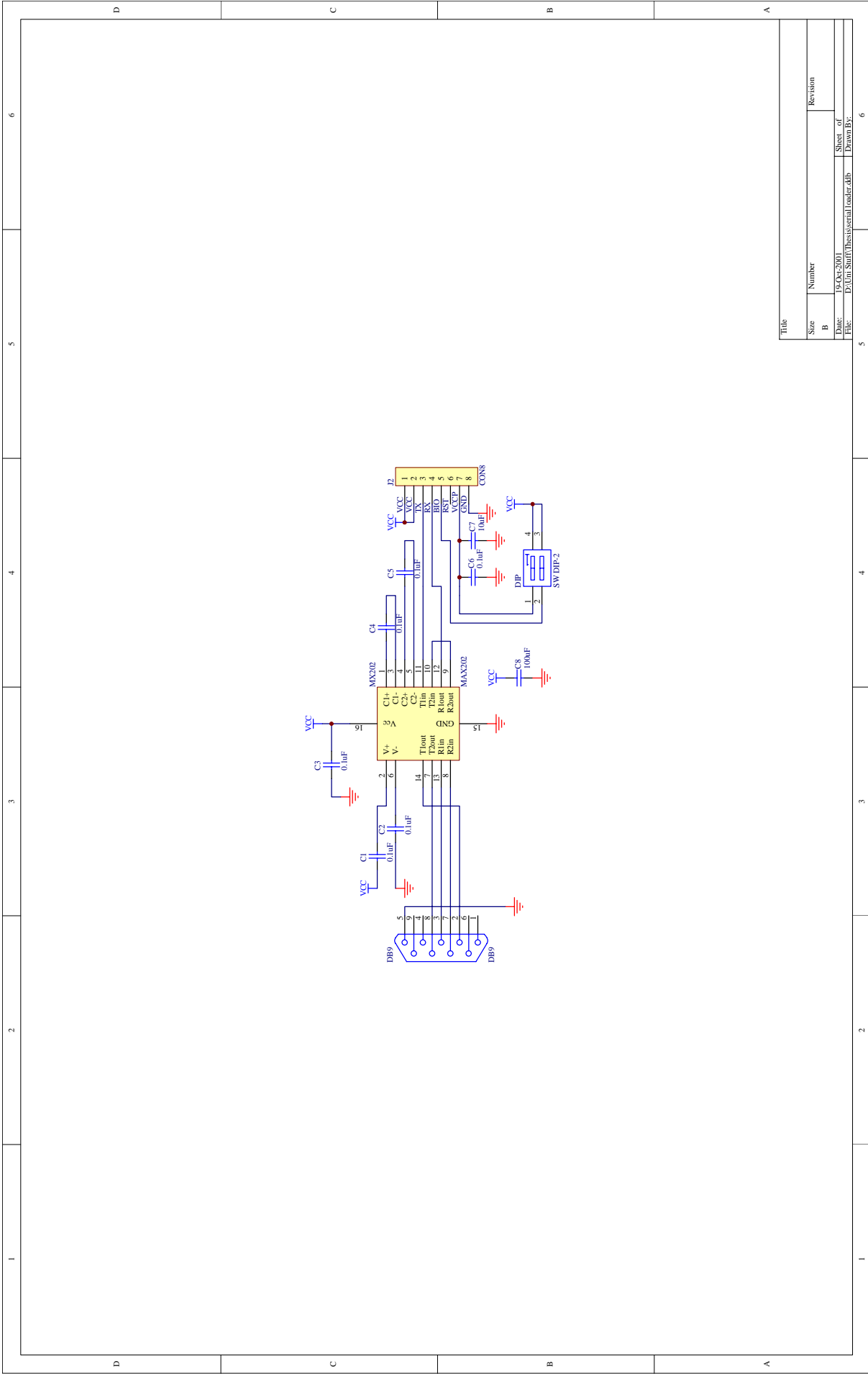
**D1 - TMS320F243A DSP**

**D2 - MAX811 Reset Regulator**

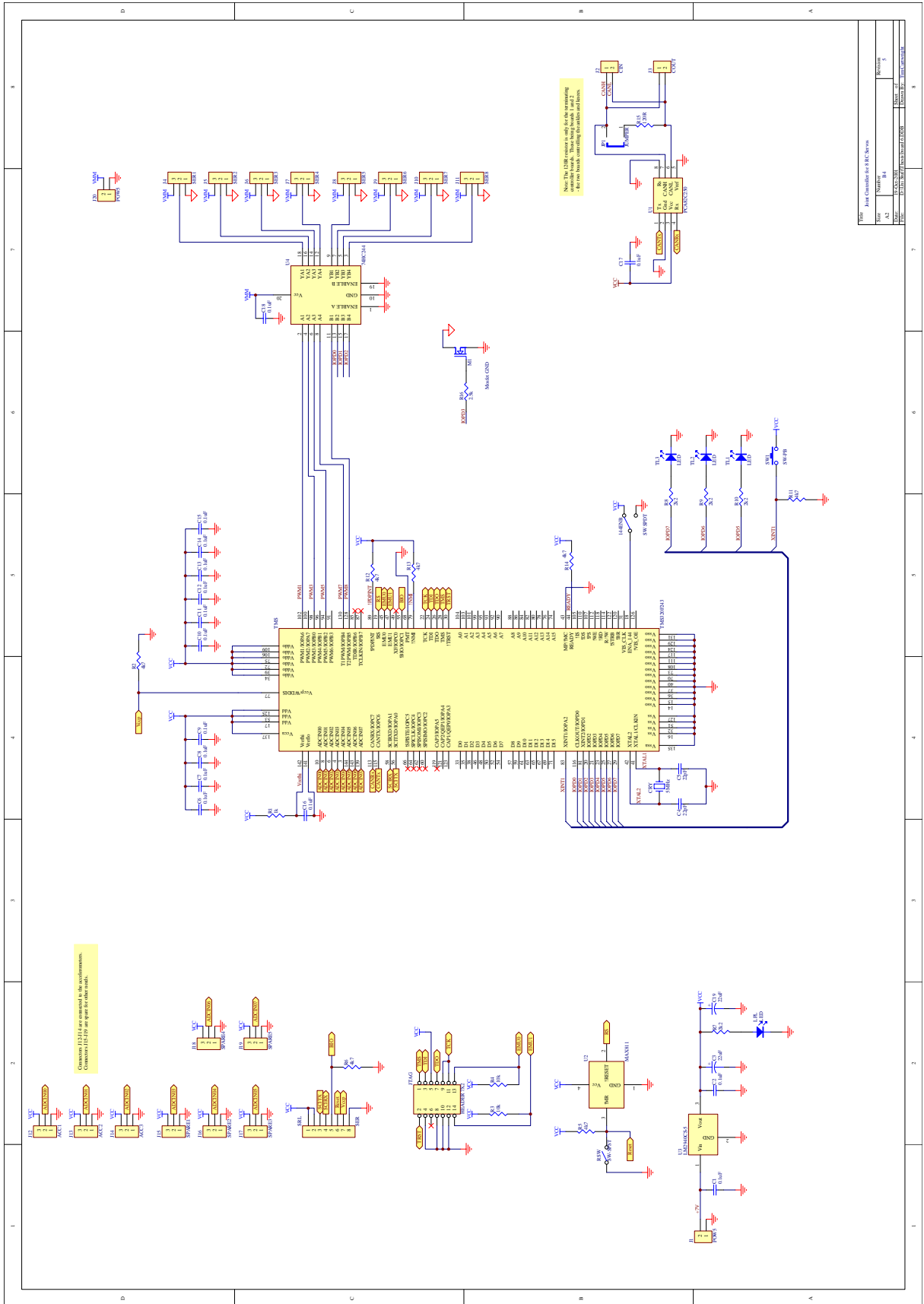
**D3 - HC244A Buffer**

## **Appendix E – Unpublished Paper G. Wyeth et al.**

**E3 - Design of an Autonomous Humanoid Robot**

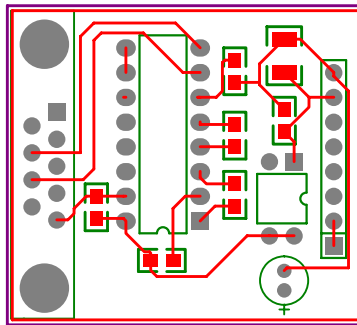


Title		Revision	
Size	Number		
B			
Date:	19-Oct-2001	Sheet of	
File:	D:\On-Staff\Theiss\serial\rs485.dtb	Drawn By:	

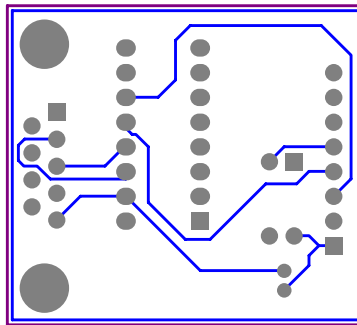




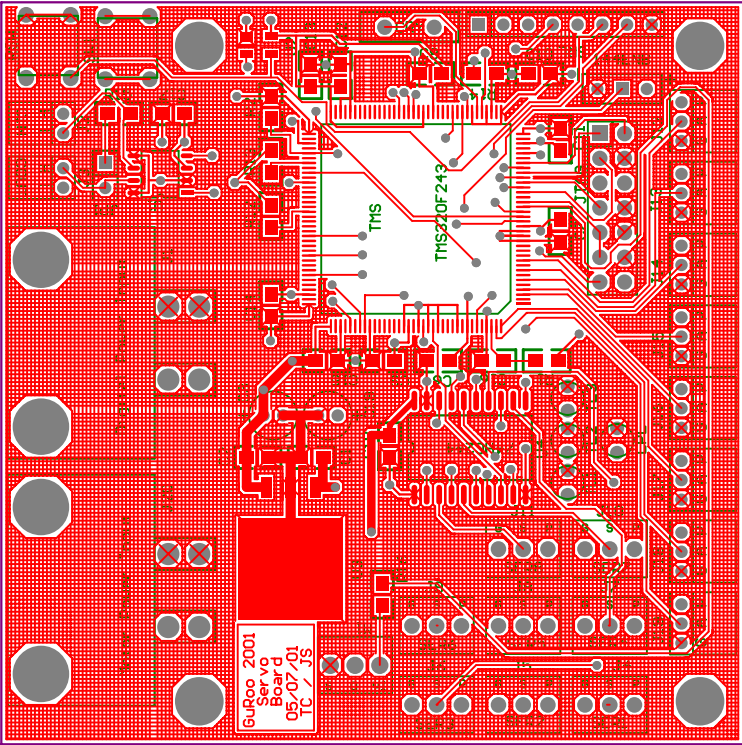
# B1 - Serial Programmer Top Layer



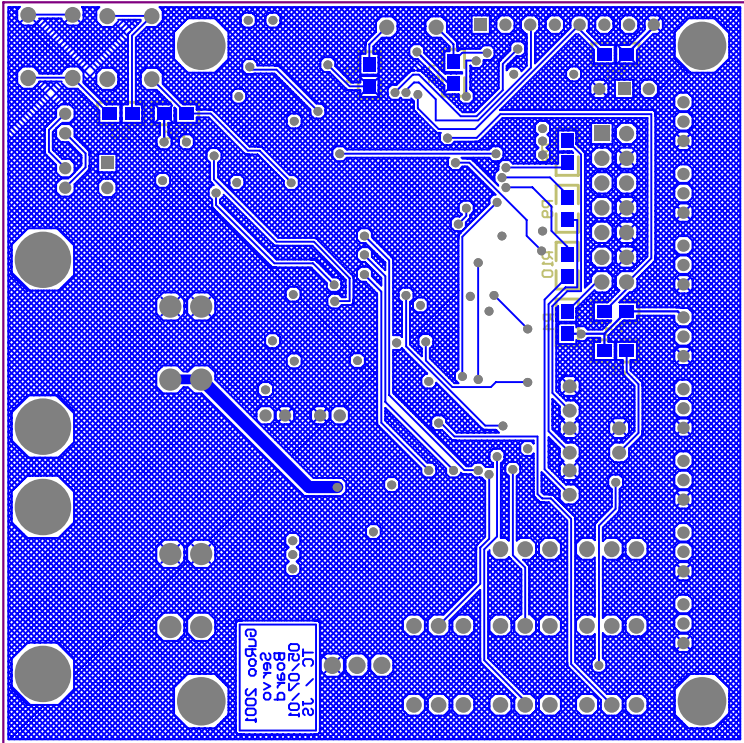
# B1 - Serial Programmer Bottom Layer



B2 - Joint Controller Top Layer



B2 - Joint Controller Bottom Layer



```

#include "F24x.h"
#include "macros.h"
#include "string.h"

void scisetaup(void);
void cansetaup(void);
void intsetaup(void);

void sciout(unsigned int output);
void sciout2(char string[50]);
void sciin(void);

void canout(unsigned int output);
void canin(void);

void wait(unsigned int time);
void c_int6(void);

void main(void)
{
    OCRA = 0xFFFF;          /* setting all IO pins as auxiliary functions */
    OCRB = 0x02FC;         /* setting all IO pins as auxiliary functions */

    WDDISABLE;
    INT_DISABLE;

    PDDATDIR = 0xFFFF;     /* All Leds On */

    scisetaup();
    cansetaup();
    intsetaup();

    while (1) {
        sciout2("Hewwwoooo World !");
        wait(100000);
    }
}

void scisetaup(void)
{
    SCICCR = 0x07;         /* Character length 8 bits. */
    SCICTL1 = 0x23;        /* RX error interrupt on, RX and TX enabled */
    SCICTL2 = 0x02;        /* RX interrupt enabled / TX disabled */
    SCIPRI = 0x60;        /* Set Priority of Interupts to Low */

    SCIHBAUD = 0x00;      /* 38461 Baud */
    SCILBAUD = 0x40;      /* BRR = 64 */
}

void cansetaup(void)
{
}

void intsetaup(void)
{
    IMR          = 0x0020;    /* INT6 enabled */
    INT_ENABLE;    /* Enable interupts */
}

void sciout(unsigned int output)
{
    unsigned int temp;

    while (!(SCICTL2&0x0080));    /* Wait for TXRDY flag */
    SCITXBUF = output;           /* Put character into Output buffer*/
}

void sciout2(char string[50])
{
    int x = 0;                   /* String sciout */
    while (string[x] != '\0') {
        sciout(string[x]);
        x++;
    }
}

void sciin(void)
{
    unsigned int input;
    PDDATDIR = 0xFF00;
}

```

```
    input = SCIRXBUF;
}

void canin(void)
{
    unsigned int input;
}

void wait(unsigned int time)
{
    unsigned int counter;
    for (counter = 0; counter < time; counter++);
}

void c_int6(void)
{
    if (SCIRXST & 0x0040){
        sciin();
    }
    IFR = 0x0020;          /* clear interrupt level 6 flag */
}
```

- High-Performance Static CMOS Technology
- Includes the TMS320C2xx Core CPU
  - Object-Compatible With the TMS320C2xx
  - Source-Code-Compatible With TMS320C25
  - Upwardly Compatible With TMS320C5x™
  - 50-ns Instruction Cycle Time
- Commercial and Industrial Temperature Available
- Memory
  - 544 Words x 16 Bits of On-Chip Data/Program Dual-Access RAM (DARAM)
  - 8K Words x 16 Bits of Flash EEPROM
  - 224K Words x 16 Bits of Total Memory Address Reach (F243 only)
- External Memory Interface (F243 only)
- Event-Manager Module
  - Eight Compare/Pulse-Width Modulation (PWM) Channels
  - Two 16-Bit General-Purpose Timers With Four Modes, Including Continuous Up and Up/Down Counting
  - Three 16-Bit Full Compare Units With Deadband
  - Three Capture Units (Two With Quadrature Encoder-Pulse Interface Capability)
- Single 10-Bit Analog-to-Digital Converter (ADC) Module With 8 Multiplexed Input Channels
- Controller Area Network (CAN) Module
- 26 Individually Programmable, Multiplexed General-Purpose I/O (GPIO) Pins
- Six Dedicated GPIO Pins (F243 only)
- Phase-Locked-Loop (PLL)-Based Clock Module
- Watchdog (WD) Timer Module
- Serial Communications Interface (SCI) Module
- 16-Bit Serial Peripheral Interface (SPI) Module
- Five External Interrupts (Power Drive Protection, Reset, NMI, and Two Maskable Interrupts)
- Three Power-Down Modes for Low-Power Operation
- Scan-Based Emulation
- Development Tools Available:
  - Texas Instruments (TI) ANSI C Compiler, Assembler/Linker, and C-Source Debugger
  - Full Range of Emulation Products
    - Self-Emulation (XDS510™)
    - Third-Party Digital Motor Control and Fuzzy-Logic Development Support
- 144-Pin LQFP PGE Package (F243)
- 68-Pin PLCC FN Package (F241)
- 64-Pin QFP PG Package (F241)

## description

The TMS320F243 and TMS320F241 devices are members of the 24x generation of digital signal processor (DSP) controllers based on the TMS320C2000™ platform of 16-bit fixed-point DSPs. The F243 is a superset of the F241. These two devices share similar core and peripherals with some exceptions. For example, the F241 does not have an external memory interface. This new family is optimized for digital motor/motion control applications. The DSP controllers combine the enhanced TMS320™ DSP family architectural design of the C2xx core CPU for low-cost, high-performance processing capabilities and several advanced peripherals optimized for motor/motion control applications. These peripherals include the event manager module, which provides general-purpose timers and PWM registers to generate PWM outputs, and a single, 10-bit analog-to-digital converter (ADC), which can perform conversion within 1 μs.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

TMS320C5x, XDS510, TMS320C2000, and TMS320 are trademarks of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



Copyright © 2000, Texas Instruments Incorporated

# TMS320F243, TMS320F241 DSP CONTROLLERS

SPRS064C – DECEMBER 1997 – REVISED SEPTEMBER 2000

## device features

Table 1 and Table 2 provide a comparison of the features of the F243 and F241. See the functional block diagram for 24x peripherals and memory.

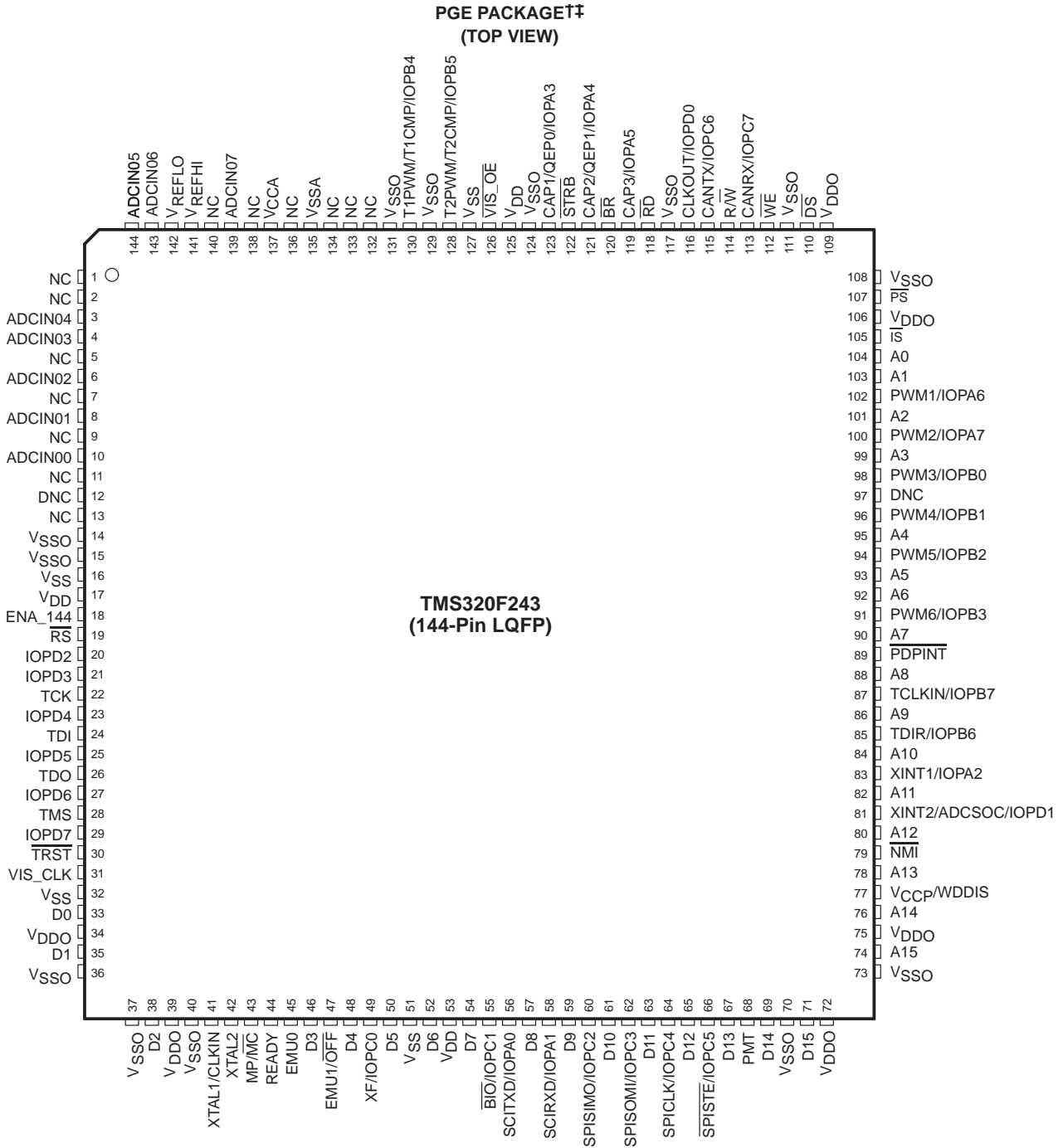
**Table 1. Hardware Features of the TMS320F24x DSP Controllers**

TMS320F24x DEVICES	ON-CHIP MEMORY (WORDS)		EXTERNAL MEMORY INTERFACE	POWER SUPPLY (V)	CYCLE TIME (ns)
	RAM				
	DATA SPACE	CONFIGURABLE DATA/PROG SPACE			
	(B1 RAM - 256 WORDS) (B2 RAM - 32 WORDS)	(B0 RAM)			
TMS320F243	288	256	√	5	50
TMS320F241			-		

**Table 2. Device Specifications of the TMS320F24x DSP Controllers**

TMS320F24x DEVICES	ON-CHIP MEMORY (WORDS)		ADC CHANNELS	PERIPHERALS		GPIO	PACKAGE TYPE PIN COUNT
	ROM	FLASH EEPROM		CAN	SPI		
	PROG	PROG					
TMS320F243	-	8K	8	√	√	32	PGE 144-PQFP
TMS320F241	-	8K	8	√	√	26	FN 68-PLCC PG 64-PQFP





† NC = No connection, DNC = Do not connect

‡ The PMT pin, number 68 in this package drawing, must be connected to ground.



# 4-Pin $\mu$ P Voltage Monitors with Manual Reset Input

MAX811/MAX812

## General Description

The MAX811/MAX812 are low-power microprocessor ( $\mu$ P) supervisory circuits used to monitor power supplies in  $\mu$ P and digital systems. They provide excellent circuit reliability and low cost by eliminating external components and adjustments when used with 5V-powered or 3V-powered circuits. The MAX811/MAX812 also provide a debounced manual reset input.

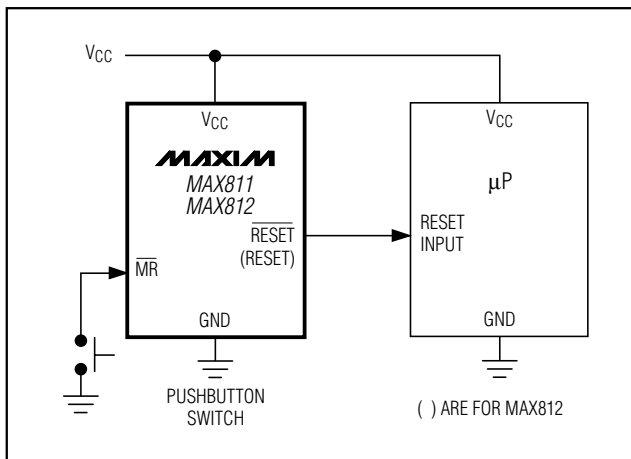
These devices perform a single function: They assert a reset signal whenever the  $V_{CC}$  supply voltage falls below a preset threshold, keeping it asserted for at least 140ms after  $V_{CC}$  has risen above the reset threshold. The only difference between the two devices is that the MAX811 has an active-low RESET output (which is guaranteed to be in the correct state for  $V_{CC}$  down to 1V), while the MAX812 has an active-high RESET output. The reset comparator is designed to ignore fast transients on  $V_{CC}$ . Reset thresholds are available for operation with a variety of supply voltages.

Low supply current makes the MAX811/MAX812 ideal for use in portable equipment. The devices come in a 4-pin SOT143 package.

## Applications

- Computers
- Controllers
- Intelligent Instruments
- Critical  $\mu$ P and  $\mu$ C Power Monitoring
- Portable/Battery-Powered Equipment

## Typical Operating Circuit



## Features

- ◆ Precision Monitoring of 3V, 3.3V, and 5V Power-Supply Voltages
- ◆ 6 $\mu$ A Supply Current
- ◆ 140ms Min Power-On Reset Pulse Width;  $\overline{\text{RESET}}$  Output (MAX811), RESET Output (MAX812)
- ◆ Guaranteed Over Temperature
- ◆ Guaranteed  $\overline{\text{RESET}}$  Valid to  $V_{CC} = 1\text{V}$  (MAX811)
- ◆ Power-Supply Transient Immunity
- ◆ No External Components
- ◆ 4-Pin SOT143 Package

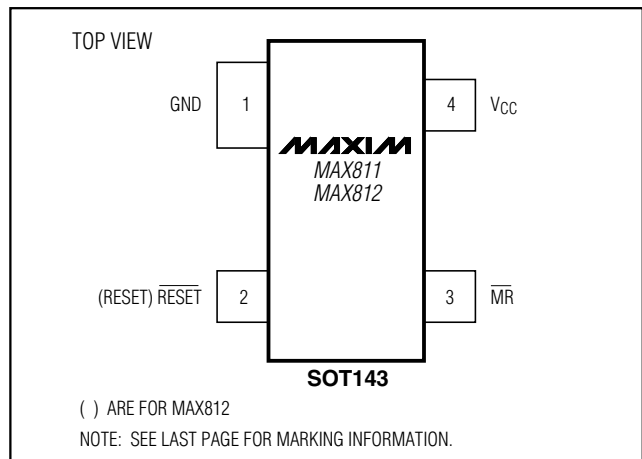
## Ordering Information

PART*	TEMP. RANGE	PIN-PACKAGE
MAX811_EUS-T	-40°C to +85°C	4 SOT143
MAX812_EUS-T	-40°C to +85°C	4 SOT143

\* This part offers a choice of five different reset threshold voltages. Select the letter corresponding to the desired nominal reset threshold voltage, and insert it into the blank to complete the part number.

RESET THRESHOLD	
SUFFIX	VOLTAGE (V)
L	4.63
M	4.38
T	3.08
S	2.93
R	2.63

## Pin Configuration



# 4-Pin $\mu$ P Voltage Monitors with Manual Reset Input

## ABSOLUTE MAXIMUM RATINGS

Terminal Voltage (with respect to GND)

V<sub>CC</sub> .....-0.3V to 6.0V

All Other Inputs.....-0.3V to (V<sub>CC</sub> + 0.3V)

Input Current, V<sub>CC</sub>,  $\overline{\text{MR}}$  .....20mA

Output Current,  $\overline{\text{RESET}}$  or  $\overline{\text{RESET}}$  .....20mA

Continuous Power Dissipation (T<sub>A</sub> = +70°C)

SOT143 (derate 4mW/°C above +70°C) 320mW

Operating Temperature Range .....-40°C to +85°C

Storage Temperature Range .....-65°C to +160°C

Lead Temperature (soldering, 10sec) .....+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## ELECTRICAL CHARACTERISTICS

(V<sub>CC</sub> = 5V for L/M versions, V<sub>CC</sub> = 3.3V for T/S versions, V<sub>CC</sub> = 3V for R version, T<sub>A</sub> = -40°C to +85°C, unless otherwise noted. Typical values are at T<sub>A</sub> = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS		
Operating Voltage Range	V <sub>CC</sub>	T <sub>A</sub> = 0°C to +70°C	1.0		5.5	V		
		T <sub>A</sub> = -40°C to +85°C	1.2					
Supply Current	I <sub>CC</sub>	MAX81_L/M, V <sub>CC</sub> = 5.5V, I <sub>OUT</sub> = 0		6	15	$\mu$ A		
		MAX81_R/S/T, V <sub>CC</sub> = 3.6V, I <sub>OUT</sub> = 0		2.7	10			
Reset Threshold	V <sub>TH</sub>	MAX81_L	T <sub>A</sub> = +25°C	4.54	4.63	4.72	V	
			T <sub>A</sub> = -40°C to +85°C	4.50		4.75		
		MAX81_M	T <sub>A</sub> = +25°C	4.30	4.38	4.46		
			T <sub>A</sub> = -40°C to +85°C	4.25		4.50		
		MAX81_T	T <sub>A</sub> = +25°C	3.03	3.08	3.14		
			T <sub>A</sub> = -40°C to +85°C	3.00		3.15		
		MAX81_S	T <sub>A</sub> = +25°C	2.88	2.93	2.98		
			T <sub>A</sub> = -40°C to +85°C	2.85		3.00		
		MAX81_R	T <sub>A</sub> = +25°C	2.58	2.63	2.68		
			T <sub>A</sub> = -40°C to +85°C	2.55		2.70		
		Reset Threshold Tempco			30			ppm/°C
		V <sub>CC</sub> to Reset Delay (Note 2)		V <sub>OD</sub> = 125mV, MAX81_L/M		40		
V <sub>OD</sub> = 125mV, MAX81_R/S/T				20				
Reset Active Timeout Period	t <sub>RP</sub>	V <sub>CC</sub> = V <sub>TH(MAX)</sub>	140		560	ms		
$\overline{\text{MR}}$ Minimum Pulse Width	t <sub>MR</sub>		10			$\mu$ s		
$\overline{\text{MR}}$ Glitch Immunity (Note 3)				100		ns		
$\overline{\text{MR}}$ to Reset Propagation Delay (Note 2)	t <sub>MD</sub>			0.5		$\mu$ s		
$\overline{\text{MR}}$ Input Threshold	V <sub>IH</sub>	V <sub>CC</sub> > V <sub>TH(MAX)</sub> , MAX81_L/M	2.3		0.8	V		
	V <sub>IL</sub>							
	V <sub>IH</sub>	V <sub>CC</sub> > V <sub>TH(MAX)</sub> , MAX81_R/S/T	0.7 × V <sub>CC</sub>					
	V <sub>IL</sub>		0.25 × V <sub>CC</sub>					
$\overline{\text{MR}}$ Pull-Up Resistance			10	20	30	k $\Omega$		
RESET Output Voltage (MAX812)	V <sub>OH</sub>	I <sub>SOURCE</sub> = 150 $\mu$ A, 1.8V < V <sub>CC</sub> < V <sub>TH(MIN)</sub>	0.8V <sub>CC</sub>		0.3	V		
	V <sub>OL</sub>	MAX812R/S/T only, I <sub>SINK</sub> = 1.2mA, V <sub>CC</sub> = V <sub>TH(MAX)</sub>						
		MAX812L/M only, I <sub>SINK</sub> = 3.2mA, V <sub>CC</sub> = V <sub>TH(MAX)</sub>			0.4			

## Octal buffer/line driver; 3-state

## 74HC/HCT244

## FEATURES

- Output capability: bus driver
- I<sub>CC</sub> category: MSI

## GENERAL DESCRIPTION

The 74HC/HCT244 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT244 are octal non-inverting buffer/line drivers with 3-state outputs. The 3-state outputs are controlled by the output enable inputs  $1\overline{OE}$  and  $2\overline{OE}$ . A HIGH on  $n\overline{OE}$  causes the outputs to assume a high impedance OFF-state. The “244” is identical to the “240” but has non-inverting outputs.

## QUICK REFERENCE DATA

GND = 0 V; T<sub>amb</sub> = 25 °C; t<sub>r</sub> = t<sub>f</sub> = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t <sub>PHL</sub> / t <sub>PLH</sub>	propagation delay 1A <sub>n</sub> to 1Y <sub>n</sub> ; 2A <sub>n</sub> to 2Y <sub>n</sub>	C <sub>L</sub> = 15 pF; V <sub>CC</sub> = 5 V	9	11	ns
C <sub>I</sub>	input capacitance		3.5	3.5	pF
C <sub>PD</sub>	power dissipation capacitance per buffer	notes 1 and 2	35	35	pF

## Notes

1. C<sub>PD</sub> is used to determine the dynamic power dissipation (P<sub>D</sub> in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f<sub>i</sub> = input frequency in MHz

f<sub>o</sub> = output frequency in MHz

∑ (C<sub>L</sub> × V<sub>CC</sub><sup>2</sup> × f<sub>o</sub>) = sum of outputs

C<sub>L</sub> = output load capacitance in pF

V<sub>CC</sub> = supply voltage in V

2. For HC the condition is V<sub>I</sub> = GND to V<sub>CC</sub>  
For HCT the condition is V<sub>I</sub> = GND to V<sub>CC</sub> – 1.5 V

## ORDERING INFORMATION

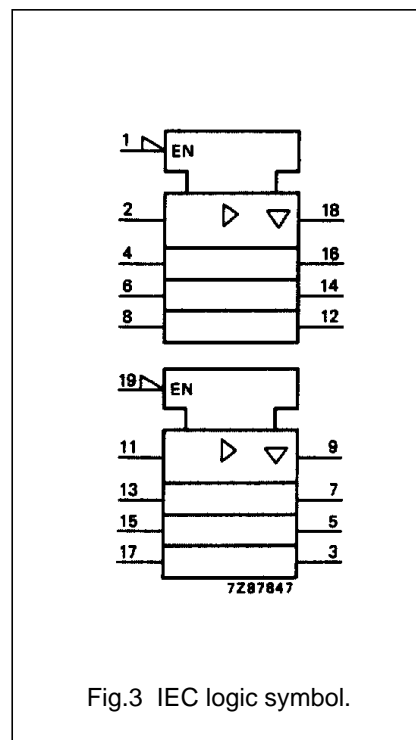
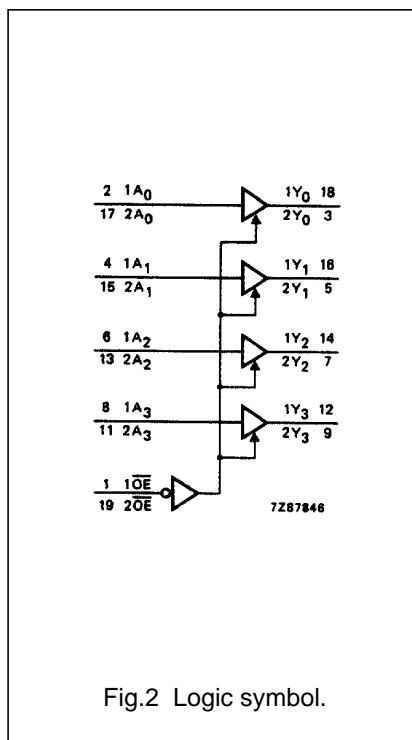
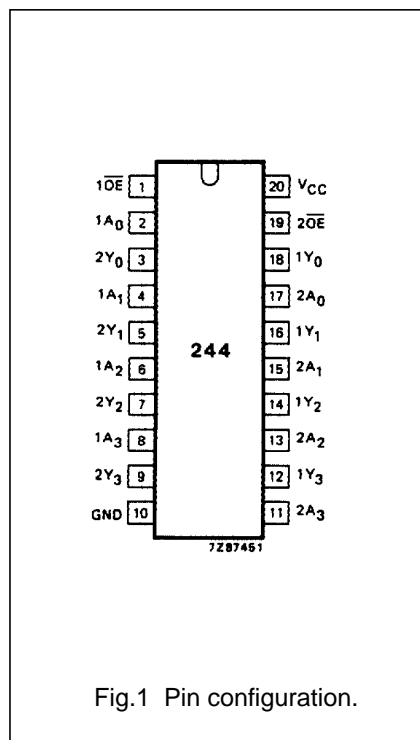
See “74HC/HCT/HCU/HCMOS Logic Package Information”.

Octal buffer/line driver; 3-state

74HC/HCT244

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1	$\overline{1OE}$	output enable input (active LOW)
2, 4, 6, 8	1A <sub>0</sub> to 1A <sub>3</sub>	data inputs
3, 5, 7, 9	2Y <sub>0</sub> to 2Y <sub>3</sub>	bus outputs
10	GND	ground (0 V)
17, 15, 13, 11	2A <sub>0</sub> to 2A <sub>3</sub>	data inputs
18, 16, 14, 12	1Y <sub>0</sub> to 1Y <sub>3</sub>	bus outputs
19	$\overline{2OE}$	output enable input (active LOW)
20	V <sub>CC</sub>	positive supply voltage



# Design of an Autonomous Humanoid Robot

Gordon Wyeth, Damien Kee, Mark Wagstaff, Nathaniel Brewer,  
Jared Stirzaker, Timothy Cartwright, Bartek Bebel  
School of Computer Science and Electrical Engineering  
University of Queensland  
St. Lucia, Queensland, 4072  
Australia

## Abstract

This paper describes the design of an autonomous humanoid robot. The robot itself is currently under construction, however the process of designing the robot has revealed much about the considerations for creating a robot with humanoid shape. The mechanical design is a complete CAD solids model, with specific motors and transmission systems selected. The electronic design of a distributed control system is also complete, along with the electronics for power and sensor processing. A high fidelity graphical simulator has been developed, providing important early feedback on critical design decisions.

## 1 Introduction

There are several reasons to build a robot with humanoid form. It has been argued that to build a machine with human like intelligence, it must be embodied in a human like body. Others argue that for humans to interact naturally with a robot, it will be easier for the humans if that robot has humanoid form. A third, and perhaps more concrete, reason for building a humanoid robot is to develop a machine that interacts naturally with human spaces. The architectural constraints on our working and living environments are based on the form and dimensions of the human body. Consider the design of stairs, cupboards and chairs; the dimensions of doorways, corridors and benches. A robot that lives and works with humans in an unmodified environment must have a form that can function with everyday objects. The only form that is guaranteed to work in all cases is the form of humanoid.

### 1.1 The *GuRoo* Project

The *GuRoo* project in the University of Queensland Robotics Laboratory aims to design and build a 1.2m tall robot with human proportions that is capable of balancing, walking, turning, crouching, and standing from a prostrate position. The target mass for the robot is 30 kg, including on-board power and computation. The robot will have active, monocular, colour vision and vision processing.

The intended challenge task for the robot is to play a game of soccer with or against human players or other humanoid robots. To complete this challenge, the robot must be able to move freely on its two legs. It requires a

vision sense that can detect the objects in a soccer game, such as the ball, the players from both teams, the goals and the boundaries. It must also be able to manipulate and kick a ball with its feet, and be robust enough to deal with legal challenges from human players. Clearly, the robot must operate in a completely autonomous fashion without support harnesses or wiring tethers.

These goals are yet to be realised for the *GuRoo* project. Currently the robot exists as a complete mechanical CAD model (see Figure 1), a complete electronic model and a high fidelity dynamic simulation. The dynamic simulation has been programmed to crouch, jump and balance. The progress to this stage has revealed much about the design considerations for a humanoid robot.

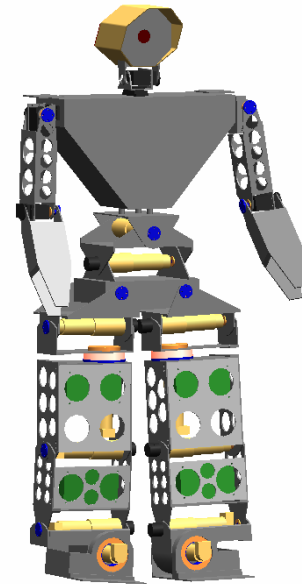


Figure 1: Full CAD model of the *GuRoo* humanoid robot.

### 1.2 Paper Overview

This section has described the motivation for building a humanoid robot, and the specific challenge that has been set for the *GuRoo* project. The subsequent section will look at other humanoid robot projects, including bipedal walking robots.

The rest of the paper describes the mechanical, electronic and software design of the *GuRoo* robot. In

particular, the paper will detail the mechanical model of the robot and a comparison to the human form, the motors and sensors, the complete electronic design, a full dynamic software simulation of the robot, the software architecture of the robot, and results for balancing and crouching in simulation.

## 2 Prior Art

### 2.1 Bipedal Walking Robots

Research into bipedal walking robots can be split into two categories: *active* and *passive*. The passive or un-powered category (for example, McGeer’s passive dynamic walker [McGeer, 1990]) is of interest as it illustrates that walking is fundamentally a dynamic problem. Passive walkers do not require actuators, sensors, or computers in order to make them move, but walk down gentle slopes generating motion by the hardware geometry. The passive walkers also illustrate the walking can be performed with very little power input.

Active walkers can further be split into two categories; those that employ the natural dynamics of specialised actuators, and those that are fully power operated. Raibert [Raibert, 1986] and later Pratt [Pratt, 1998] have shown some impressive feats of walking and gymnastic ability in robots that have the capacity for energy storage in the actuator. These robots have been shown to have robust and stable performance from relatively simple control mechanisms.

The alternate approach is to control the joints through pre-specified trajectories to a known “good” gait pattern (for example, [Golden, 1990]). This is a simple approach, but lacks robustness to disturbances. This approach becomes more complex when additional layers are added to provide adjustments to the gait for disturbance. Controlling a fully powered biped in a manner that depends on the dynamic model is complicated by the complex dynamic equations for the robot’s motion. Yamaguchi et al. [Yamaguchi, 1998] moved a dynamic torso with significant mass through 2 DOF to keep the Zero Moment Point (ZMP) within the polygon of the support foot. This approach contributed to successful control of the robot, but produces an awkward gait.

### 2.2 Bipedal Walking Humanoid Robots

There are few examples of autonomous biped walkers that resemble the structure of a human. The Honda company biped robots, P2 and P3 are two of the few examples of such robots [Hirai, 1998]. P3 can walk on level ground, walk up and down stairs, turn, balance, and push objects. The robot is completely electrically and mechanically autonomous. The Sony SDR-3X robot is another example with similar capabilities, although details of the design are yet to be published.

## 3 Mechanics

The mechanical design of the humanoid requires careful and complex tradeoffs between form, function, power, weight, cost and manufacturability. For example, in terms of form, the robot should conform to the proportions of a

1.2m tall human. However, retaining the exact proportions compromises the design in terms of the selection of actuation and mechanical power transmission systems. Affordable motors that conform to the dimensional restrictions have insufficient power for the robot to walk or crouch. This section describes the final mechanical design and how the balance between conflicting design requirements has been achieved.

### 3.1 Proportions

The target proportions for the robot are based on biomechanical data of the human form. Figure 2 shows the proportions of the frontal plane dimensions of a 50<sup>th</sup> percentile male based on data from a United States survey [Dempster, 1965]. The dimensions shown in millimetres indicate the appropriate sizes of anatomical features when scaled to a total height of 1200 mm against the comparable dimensions on GuRoo.

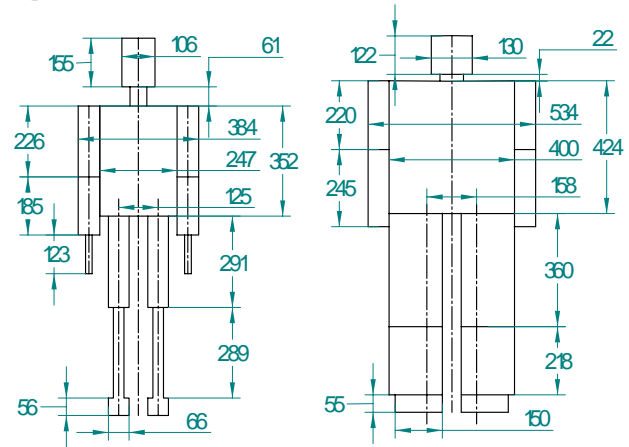


Figure 2: The proportions of typical human anatomy compared to the matching proportions of GuRoo’s anatomy. The dimensions indicate the sizes for a human scaled to 1.2m in height.

By comparison, GuRoo is somewhat thickset in the legs, as was dictated by the form of the chosen actuators (see Section 3.3). The spacing between the hips and ankles has been retained, rather than placing the hips and ankles along the frontal centreline of each leg. Our simulation studies showed that the required torques around the roll axes of the hips and ankles becomes excessive if the hips and ankles are spaced too far apart (see Section 5.3).

The body and upper leg of GuRoo are somewhat longer than the counterparts in the human model. This is due to the chain of actuators required for three degrees of freedom in the waist and hips respectively (see Section 3.2). Consequently, the lower leg and the neck and head are shorter to compensate. The overall effect is still convincingly human-like in shape.

The changes in volume required to house the actuators, as well as the mass of the actuators themselves have an effect on the mass distribution. Table 1 shows the mass distribution of GuRoo compared to that of a human. The most notable exception is that the shin and foot are much heavier in GuRoo than the human counterpart, due to the mass of the powerful actuators required in the ankle. The arms are significantly lighter than the human

counterpart, as they are significantly inferior in power and do not have hands. GuRoo's mass distribution is closer to the human distribution than either MIT's active bipedal walker [Paluska, 2000], or McGeer's passive dynamic bipedal walker.

Table 1: Comparison of GuRoo mass distribution with human mass distribution, and with the mass distribution of MIT's M2 bipedal walker and McGeer's passive dynamic walker.

Body Component	GuRoo mass (kg)	GuRoo	Human	M2	PDW
Head and Upper torso	7.3	24%	31%	0%	0%
Abdomen and Hips	9.1	30%	27%	51%	50%
Thigh	5.8	19%	20%	22%	30%
Shin and Foot	6.4	21%	12%	27%	20%
Arm	1.9	6%	10%	0%	0%
Total	30.5				

The other notable point from Table 1 is the total mass of the robot. A 1.2 m tall human would typically be a child approaching his or her 7<sup>th</sup> birthday, with a 50<sup>th</sup> percentile mass of 23 kg. A child with mass of 30.5 kg at the same age would be in 97<sup>th</sup> percentile, indicating that GuRoo is somewhat overweight.

### 3.2 Architecture

The extent to which human joint function can be replicated is another key factor in robot design. Figure 3 shows the degrees of freedom contained in each joint area of the robot. In the cases where there are multiple degrees of freedom (for example, the hip) the joints are implemented sequentially through short links rather than as spherical joints. Other key differences to the human form are the lack of a continuous flexible spine, and the lack of a yaw axis in the ankle. Another point to note is that the roll and pitch axes of the ankle are orthogonal, whereas the human ankle has an angle of about 64° between the roll and pitch axes.

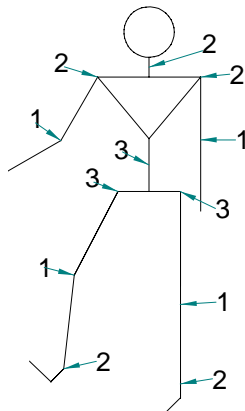


Figure 3: The location of the joints in GuRoo, indicating the degrees of freedom in each joint.

### 3.3 Motor Choice

The key element in driving the mechanical design has been the choice of actuator. The robot has 23 joints in total. The legs and abdomen contain 15 joints that are required to produce significant mechanical power, most generally with large torques and relatively low speeds. The other 8 joints drive the head and neck assembly, and the arms. The torque and speed requirements are significantly less. Factors of cost, weight and availability limited the choice of actuators to rotary DC motors

The 15 high power joints all use the same motor-gearbox combination. The motor is a Maxon RE 36 wound for a nominal voltage of 32V. This motor can provide 88.5 mNm of torque continuously, with a matching current consumption of 1.99 A. The motor has a maximum permissible speed of 8200 RPM. The gearbox has a reduction of 156, with an efficiency of 72%. The maximum continuous generated output torque is 10 Nm, with a maximum output speed of 51 RPM, or 5.3 rad/s. The thermal limits of the motor permit intermittent output torque of up to 19Nm. Each motor is fitted with an optical encoder for position and velocity feedback. The total mass of the motor/gearbox/encoder unit is 0.85 kg.

The 8 low power joints are Hi-Tec RC servo motors model HS705-MG. These motors have an integrated gearbox and have rated output torque to 1.4 Nm, at speeds of 5.2 rad/s. These also have potentiometer feedback and built-in control and power electronics. They require 6V power, and a pulse width modulated signal to indicate desired position. The mass of each unit is 0.125 kg.

## 4 Electronics

A distributed control network controls the robot, with a central computing hub that sets the goals for the robot, processes the sensor information, and provides coordination targets for the joints. The joints have their own control processors that act in groups to maintain global stability, while also operating individually to provide local motor control. The distributed system is connected by a CAN network. In addition, the robot requires various sensor amplifiers and power conversion circuits.

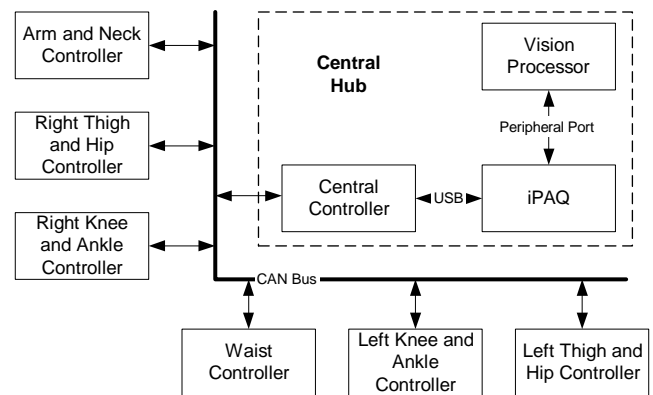


Figure 4: Block diagram of the distributed control system.



## 4.1 Computing

### 4.1.1 Central Hub

The central control of the robot derives from a hub of three heterogeneous microprocessors that provide coordination between joints, integrate sensor information, and process the vision input. This hub also provides communication to the outside world through user interfaces and communication peripherals.

The primary component of the central controller is an iPAQ pocket pc from Compaq. The iPAQ features a 208 MHz StrongARM microcontroller, 32 Mb of RAM and a 320 x 240 colour screen. The screen is touch sensitive allowing stylus input of text and graphics. The iPAQ has 16 Mb of Flash ROM to store the operating system. The iPAQ in the GuRoo operates with Windows CE. As well as the touch screen interface, the iPAQ is equipped with a speaker and microphone, a joystick, and four push-buttons. It has an infra-red interface for external communication.

The second component of the central hub is a TMS320F243 microcontroller that acts as an adapter and filter for the robot's internal CAN network (see Section 4.1.3). The microcontroller communicates with the robot's distributed control system through the CAN network, and to the iPAQ through the iPAQ's USB serial communication port. The microcontroller also manages the power supply (see Section 4.2.3) providing centralised control of the robot power supply in the event of system failure. This microcontroller is the same device used in the joint controllers (see Section 4.1.2).

The final component of the central is the vision processing board. This board has been developed for the ViperRoos robot soccer team [Chang, 2001] and features a 200 MHz Hitachi Super-H SH4 microcontroller, an FPGA-based programmable camera and bus adapter, 16 Mb of RAM, 8 Mb of flash ROM, and 512 kb of fast SRAM for video caching. The board interfaces to the 100 pin parallel peripheral bus on the iPAQ to provide real time visual display on the iPAQ's colour screen. The vision input comes from a custom digital CMOS camera, based around the OV7620 camera chip from OmniVision, which can provide 640 x 480 images at up to 25 fps. The camera can provide data in YUV or RGB formats, and can be programmed to only send data from selected areas of the sense region.

### 4.1.2 Joint Controllers

The TMS320F24x series is a 32 bit DSP designed for motor control. The availability of the Control Area Network (CAN) module in this series, along with bootloader programmable internal Flash memory makes the device particularly attractive for this application. Furthermore the device features 8k words of internal flash memory, 8 PWM channels with deadband generation, quadrature input circuitry, an 8 channel 10 bit analog to digital converter with a conversion time of 800ns, a power drive protection external interrupt, and a 50ns instruction time. The TMS320F241 from Texas Instruments operates at 20MHz, and can read the A/D converter, calculating a PID control law, current limit, and generate the required PWM output, in under 10  $\mu$ s [Wyeth, 2001]. In this application, we use the TMS320F243, which has an

external bus that is used for attaching additional sensor interfaces. Five controller boards control the 15 high power motors, each board controlling three motors. A sixth controller board controls the eight RC servo motors.

### 4.1.3 Internal Network

The CAN bus is a highly reliable standard developed by Robert Bosch GmbH for use in the automotive environment. It is a multi-master system, with sophisticated error checking and arbitration, so that any high priority message will always get through first without corruption by other messages. All data contained in each packet (up to eight bytes) is also checked with a Cyclic Redundancy Check (CRC) error-checking scheme that can correct up to five random errors, and will be automatically retransmitted if not correct. The network operates at up to 1 Mbit/sec.

## 4.2 Power

### 4.2.1 Drive Power Electronics

The drive power electronics is based on a switch mode power stage, requiring only a single supply rail and having an efficiency over 90%. This efficiency results in several advantages such as small size, lower cost power devices and less heatsinking. The H-Bridge channels are driven from separate PWM outputs of the DSP, allowing the deadband features of the PWM peripheral to be used, along with the immediate (<12ns) shutdown of these pins in the event of a fault which triggers the Power Drive Protect Interrupt (PDPInt) pin on the DSP.

A integrated solution was chosen for this design – the SGS-Thomson L6203. This device uses low on-resistance and fast switching MOSFETs, to give maximum efficiency and best control. The voltage limit of the devices is 48V, and the total continuous RMS current limit is 4A. This is a good match to the chosen motors and batteries. The total on-resistance of the power devices is 0.3 $\Omega$ . The cost of the device is low, compared to a discrete solution, and the volume and mass of the electronics is minimised by the choice of an integrated solution.

### 4.2.2 Battery Packs

The power for the 15 high power motors is provided by 4 x 1.5Ah 42V NiCd packs. These packs are effectively paralleled to a common bus (see Section 4.2.3). The packs are chosen to give 20 minutes of continuous operation. The power for the 8 low power motors is derived from a single 3Ah 7.2 V NiCd battery pack. The power for the control electronics is derived from a second single 3Ah 7.2V NiCd pack. The voltage from this pack is distributed to the various boards that require power where it is regulated locally.

### 4.2.3 Power Regulation

Connecting NiCd batteries in parallel can be extremely hazardous to the life of the batteries. Uneven charging and discharging characteristics between packs can lead to uneven load sharing and high current circulation between packs. The power from each pack is controlled through switch mode buck converters to provide even current sharing between packs, providing a voltage bus at marginally below the lowest battery voltage.

## 4.3 Sensing

### 4.3.1 Joint Sensing

Current sensing is performed in the high power joints by a  $0.01\Omega$  resistance in the ground leg of the H-Bridge. The voltage from these sense resistors is amplified by differential amplifiers and measured by the ADC. Current is also checked against a screwdriver adjustable hard limit that is used to trigger the Power Drive Protect interrupt. The position feedback from the encoders on the high power joints provides a count on every edge of both quadrature channels. This provides 2000 counts per motor revolution from the 500 count encoder wheels. In addition, each DSP can measure the bus voltage, and the temperatures of the MOSFETs and motors.

### 4.3.2 Motion Sensing

In addition to the sensing in each joint, and of course the visual feedback, the robot features 2 x 2-axis accelerometers to provide information about the torso's dynamic behaviour and the relationship to the vertical gravity force. While it is impossible to resolve the motion components of the body's acceleration from the effects of gravity, these sensors may be able to provide information with regard to disturbances while walking – playing a similar role to the human middle ear.

Provision has also been made for the contact switches in the feet and in the joints. These switches may prove useful for determining when contact is made with the ground, or initialising joints at robot start up.

## 5 Software

The software consists of four main entities: the global movement generation code, the local motor control, the low-level code of the robot, and the simulator. The software is organised to provide a standard interface to both the low-level code on the robot and the simulator. This means that the software developed in simulation can be simply re-compiled to operate on the real robot. Consequently, the robot needs a number of standard interface calls that are used for both the robot and the simulator. Figure 5 shows modularisation of the software, and the common interfaces.

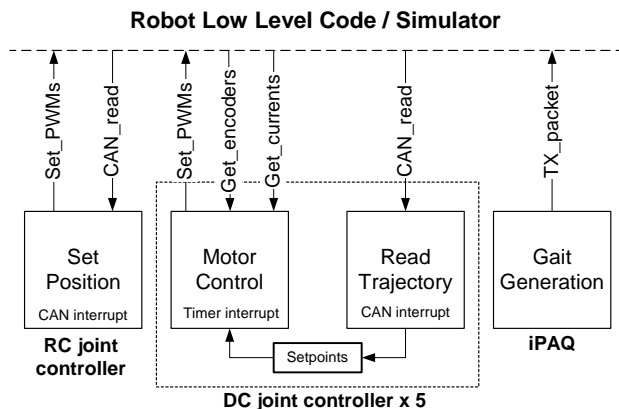


Figure 5: Block diagram of common software modules and the interface used to both the real robot and the simulator.

## 5.1 Simulator

At present, all evaluations of the robot have taken place in a high fidelity dynamic simulator. The simulator is based on the *DynaMechs* project [McMillan, 1995]. *DynaMechs* is an object-oriented, open source code library that provides full dynamic simulation for tree-structured robots having a star topology. The algorithms are capable of simulating fixed and mobile bases. The library is based on efficient recursive algorithms for the dynamic calculations, and provides graphical display of the robot in an OpenGL environment.

The simulator uses the *DynaMechs* package as the core, with additions to simulate specific features of the robot such as the DC motors and motor drives, the RC servos, the sensors, the heterogeneous processing environment and the CAN network. These additions provide an identical interface between the dynamic graphical simulation and the controller and gait generation code. The parameters for the simulator are derived from the CAD models and the data sheets from known components. These parameters include the modified Denavit-Hartenberg parameters that describe the robot topology, the tensor matrices of the links and the various motor and gearbox characteristics associated with each joint. The surface data from the CAD model is also imported to the simulator for the graphical display.

The simulator uses an integration step size of  $500\mu\text{s}$  and updates the graphical display every 5ms of simulated time. When running on 1.5 GHz Pentium 4 under Windows 2000, the simulation updates all 23 joints at a very useable 40% of real time speed.

## 5.2 Joint Controller Software

For the high power DC motor joints, the simulator provides the programmer with readings from the encoders and the current sensors, based on the velocities and torques from the dynamic equations. In the case of the RC servos, the simulator updates the position of the joints based on a PD model with a limited slew rate. The programmer must supply the simulator with PWM values for the motors to provide the control. The simulator provides fake interrupts to simulate the real events that are the basis of the control software.

There are two types of joint controller boards used in the robot – five controller boards control the fifteen high power motors and one controller controls the eight low power motors. The controller software for the low power motors is a single interrupt routine that is triggered by the arrival of a CAN packet addressed to the controller's mailbox. The routine reads the CAN mailbox for the change in position sent by the gait generation routine. The PWM duty cycle that controls the position of the RC servos is varied accordingly.

The control loop for the high power controllers has two interrupt routines. As for the low power controller, an interrupt is executed upon receipt of trajectory data in the CAN mailbox. The data is used to set the velocity setpoints for the motor control routine. There is also a periodic interrupt every  $500\mu\text{s}$  to run the motor control software. The motor control routine compares the error between velocity setpoint and the encoder reading and generates a PWM value for the motor based on a Proportional-Integral control law. The routine also checks

the motor current against the current limits, and adjusts the PWM value to prevent over-current situations.

### 5.3 Motion Generation Software

To this point, the software for motion generation has been used to test the designed geometries and chosen motors in the simulator. The software uses only local joint feedback; it does not use feedback from the joint sensors in a global sense or use the motion sensors to modify the motion to maintain balance. The tests are run without current limiting in the local control loop to evaluate worst-case performance.

The first test motion is a crouch with a return to the standing position. This test has been designed to evaluate the required torques in the pitch joints of hip, knee and ankle. The worst-case results for the knee joint are shown in Figure 6. The second test motion is a lean to balance over one leg, designed to evaluate the required torques in the roll joints of hip and ankle. The joints are driven according to the following equations. The worst-case results for the ankle are shown in Figure 7. In both of these worst cases, the current consumption only briefly exceeds the continuous current rating, and the motor stays within thermal limits.

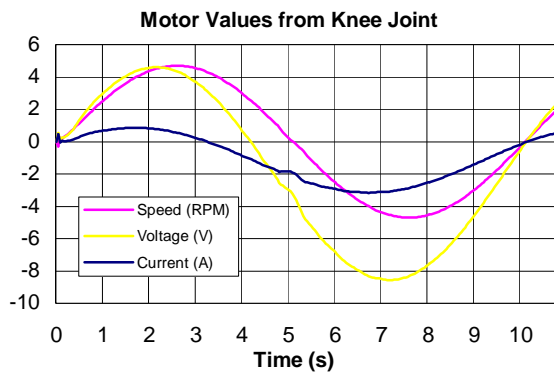


Figure 6: Simulation results for knee motor during a squatting movement. The movement cycle time is 10 seconds.

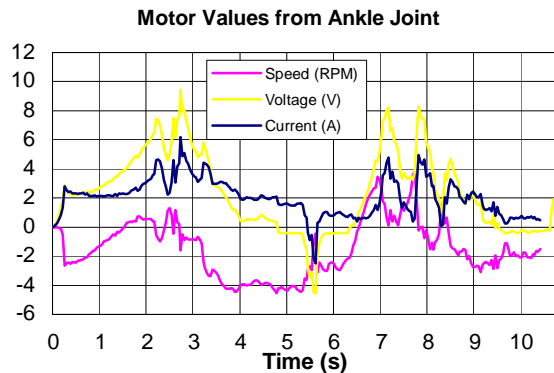


Figure 7: Simulation results for ankle motor during a balancing movement. The movement cycle time is 10 seconds.

## 6 Conclusions

This paper has illustrated the design of a practical,

affordable, autonomous, humanoid robot. The robot is well proportioned in relation to the human form, with most of the major degrees of freedom of the human body implemented. The robot design has a distributed control design with processors dedicated to each of the key roles around the robot. Investigations of the CAD design using a high fidelity simulation have shown that robot is capable of crouching and balancing.

[**Note for reviewers:** This project involves a large team who intend to have the real robot constructed and walking by September. The final paper will have further results, and the conference presentation is likely to feature a video, and possibly the robot itself.]

## References

- [Chang, 2001] M. Chang, B. Browning and G. Wyeth. ViperRoos 2000. RoboCup-2000: Robot Soccer World Cup IV. Lecture Notes in Artificial Intelligence 2019. Springer Verlag, Berlin, 2001.
- [Dempster, 1965] W.T.Dempster and G.Gaughran. Properties of body segments based on size and weight. American Journal of Anatomy, 1965.
- [Golden, 1990] J. A. Golden and Y. F. Zheng. Gait Synthesis For The SD-2 Biped Robot To Climb Stairs. International Journal of Robotics and Automation 5(4). Pages 149-159, 1990.
- [Hirai, 1998] K. Hirai, M. Hirose, Y. Haikawa, and Takenaka. The Development of Honda Humanoid Robot. IEEE Conference on Robotics and Automation, 1998.
- [Hodgins, 1995] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating Human Athletics. In Computer Graphics, (Siggraph 1995).
- [McMillan, 1995] S. McMillan, Computational Dynamics for Robotic Systems on Land and Underwater, PhD Thesis, Ohio State University, 1995.
- [McGeer, 1990] T. McGeer. Passive Dynamic Walking. International Journal of Robotics Research, 9(2):62-82, 1990.
- [Paluska, 2000] D.J. Paluska, Design of a Humanoid Biped for Walking Research, Masters Thesis, MIT, 2000.
- [Pratt, 1998] J. Pratt and G. Pratt. Intuitive Control of a Planar Bipedal Walking Robot. IEEE Conference on Robotics and Automation, 1998.
- [Raibert, 1986] M. H. Raibert. Legged Robots that Balance. MIT Press, Cambridge, MA, 1986.
- [Wyeth, 2001] Wyeth G.F., Kennedy J. and Lillywhite J. (2000) Distributed Digital Control of a Robot Arm, Proceedings of the Australian Conference on Robotics and Automation (ACRA 2000), August 30 - September 1, Melbourne.
- [Yamaguchi, 1998] J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi. Development of a Bipedal Humanoid Robot Having Antagonistic Driven Joints and Three DOF Trunk. Proceedings of the 1998 IEEE/RSJ Conference.