CHAPTER 3
ELECTRONIC SYSTEMS

3.1 Overview

While much time and effort was devoted to the mechanical structure of *Pneuman*, it would be a lifeless statue without the electronic systems. *Pneuman's* electrical system consists of four sealed lead-acid batteries, a power distribution block, a regulated power supply, an embedded computer, control electronics, actuators, and sensors. Each sub-system will be discussed, and a block diagram of the overall system is shown in Figure 3-1.

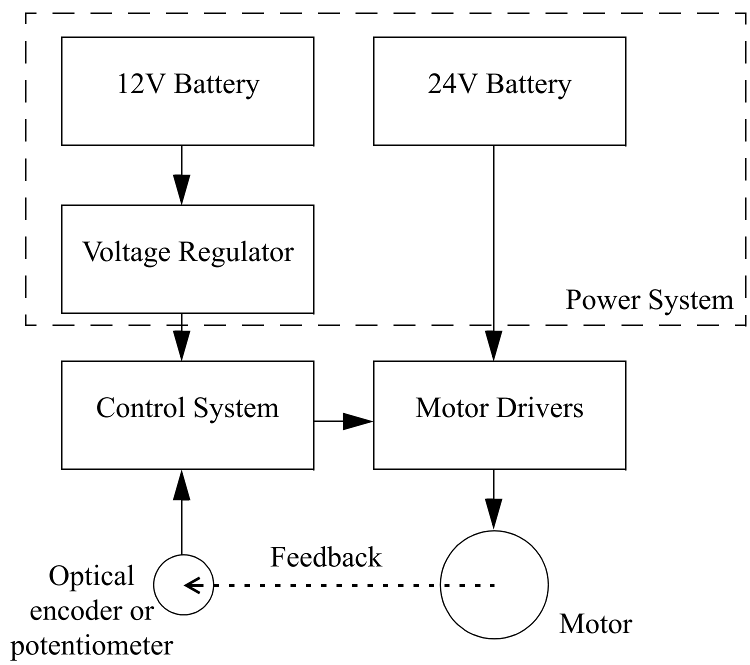

Figure 3-1: Block diagram of *Pneuman's* electronics and control systems.

3.2 Power System

Sealed lead-acid batteries were chosen due to their high capacity and low cost. Two 12V, 12Ah batteries are arranged in parallel for 12V@24Ah and two are arranged in series for 24V@12Ah. These four batteries provide *Pneuman* with enough power to operate autonomously for approximately 30 minutes. The 12V system supplies power to the computer voltage regulator and the 24V system powers all of the actuators. See Figure 3-2.

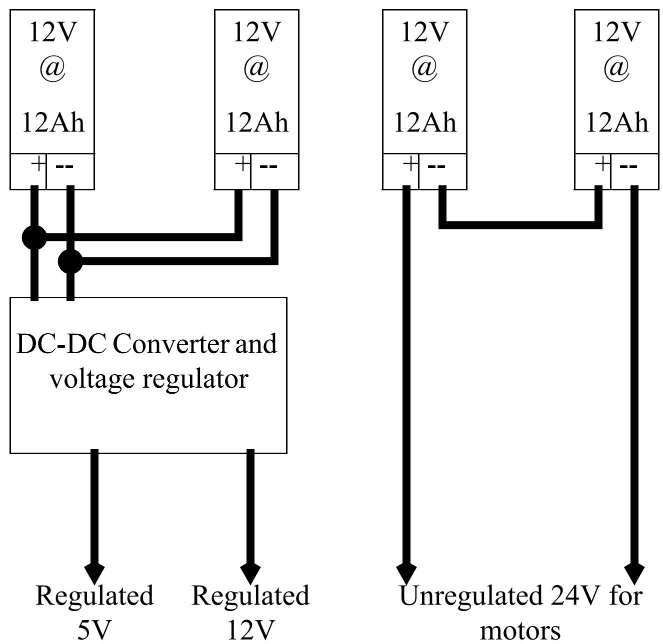

Figure 3-2: *Pneuman's* battery and power system.

3.3 Computer System

3.3.1 Hardware

Pneuman's computer is a JUMPtec Intel Pentium with a 166 MHz internal clock and 32 kB of write-back-cache. It has 128 MB of SDRAM, an 18 GB hard disk drive, a 100 BaseT ethernet connection, two serial ports, one parallel port, a floppy driver interface, a real time clock, and 128 kB FLASH BIOS. The computer is in a pc/104 form factor, making it ideal for an embedded system.

The pc/104 bus allows for easy expansion. A voice synthesizer module from RC Systems, Inc. connects to the pc/104 bus and provides *Pneuman* with the ability to speak an infinite number of words. Additionally, wireless communication is possible with the use of a pc/104 to PCMCIA wireless network card adapter. Other modules using the pc/104 bus include the PWM module and the PID module. These two modules form the main components of the digital control systems on *Pneuman*.

3.3.2 Software

The operating system is Mandrake Linux 8.1. Software was developed in the C programming language. The code is modular; each module is a separate process that communicates via a shared memory space. The main executable is a process manager that initializes memory space for the other processes and executes the requested modules. The current modules display the text user interface, generate the timer signals, and execute the control algorithms. See Figure 3-3.

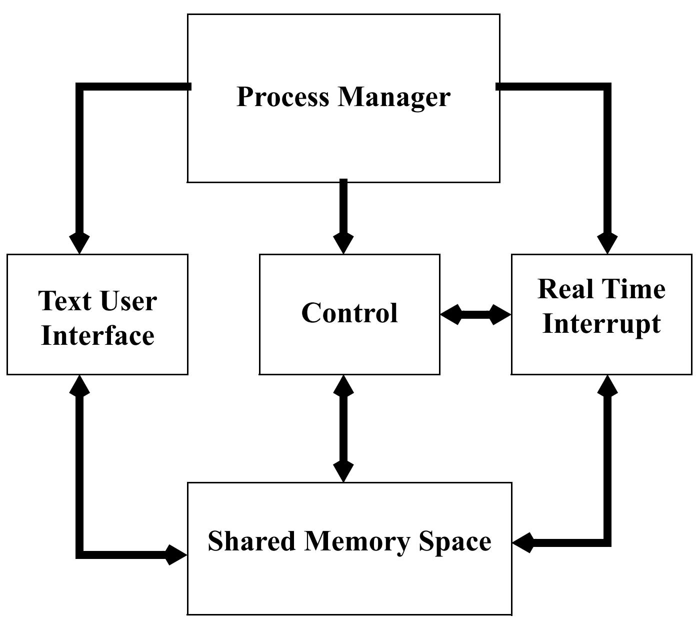

Figure 3-3: Software structure.

The results from the processes are then sent to the other modules via a shared memory space. The movement control module keeps precise records of the current and desired states of all the actuators. This data is used to determine what control is necessary to move to a desired position. The text-based user interface module allows the user of the system to change the parameters for *Pneuman*.

3.4 Pulse Width Modulation (PWM) System

All of the actuators on *Pneuman* are brushed direct current (DC) motors. Their torque is controlled by pulse width modulating their current. The control signals are generated by custom pc/104 cards. Each card has three main systems, including PWM components, digital input/output (IO) components, and an analog to digital converter. The overall is to provide a complete PID controller for each motor, implemented in software. This will be possible by using the analog inputs for angle position sensors, and controlling the motors via a motor driver board with PWM and direction signals.

The first system is responsible for generating PWM using three standard 8254 programmable timers. Each timer chip contains three individual timers, for a total of nine timers on each PC/104 board. Each timer has a count register and an output pin associated with it. When the count register reads zero, an event on the corresponding timer pin may occur. An event may involve the output going high, low, changing from its current state, or nothing at all. One of the timers is set to operate as a real-time interrupt (RTI) providing a signal that corresponds to the period of the PWM signals. Note that all eight of the PWM signals generated must have the same period. This RTI signal is connected to the trigger inputs of all the other timers. Furthermore, the remaining eight timers are operating in “one-shot” mode. This means that once the trigger is asserted, the outputs of these timers are asserted until their count registers have reached zero, thereby de-asserting the output. Additionally, the values in the count registers are loaded with different values via the PC/104 bus, thereby changing the time that the outputs remain high. Thus, this may be used in combination with the RTI as mentioned above for hardware PWM [27]. See figure 3-4.

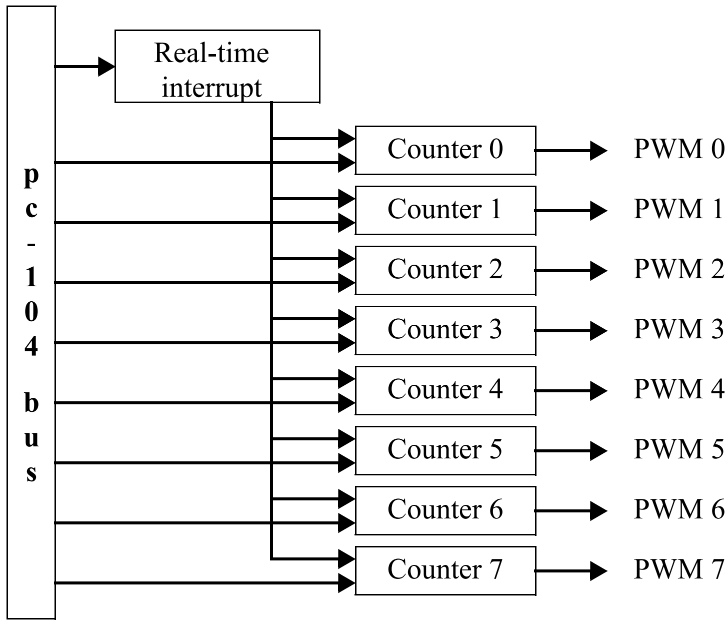

Figure 3-4: Pulse width modulation system.

The next system provides eight digital input/output connections (IO's) for the direction control of each motor, eight outputs to control the analog to digital converter, and eight inputs to read the analog values. This system uses a standard 8255 parallel peripheral interface (PPI) IC. The final component of the board, the analog to digital conversion system, uses an Analog Devices ADC0808 IC. This particular IC provides eight input channels as well as eight bits of resolution for each channel [27]. See figure 3-5. An image of the pc/104 card is shown in Figure 3-6.

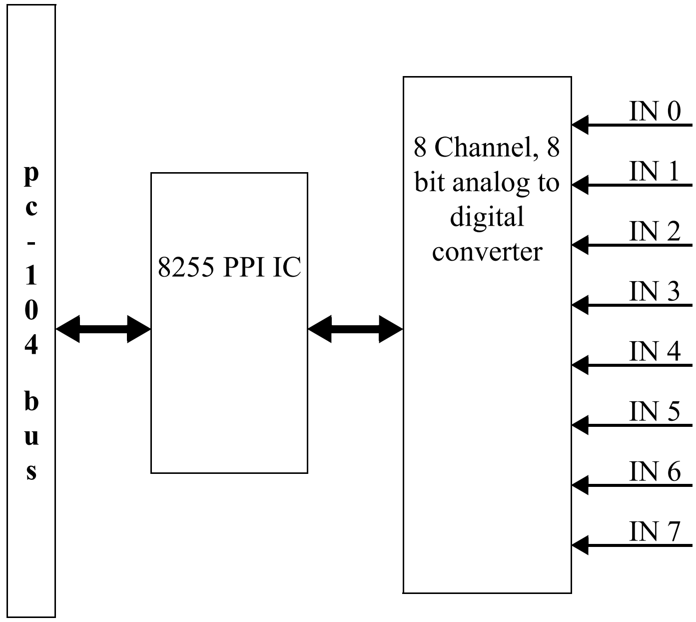

Figure 3-5: Analog to digital conversion system.

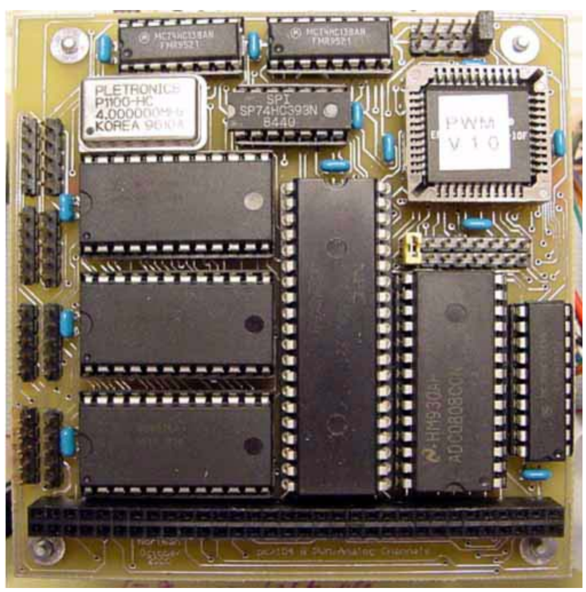

Figure 3-6: Image of the PWM and A to D pc/104 card.

3.5 Hardware-Implemented Control System

The above mentioned PWM boards are ideal for a closed loop position control system utilizing a potentiometer for feedback. However, the drive wheels use an incremental quadrature optical encoder for feedback. The encoders allow 360º of rotation, as required for a drive wheel, and the incremental count permits significant wheel travel before counter overflow. A different system was required to control the velocity and position of the drive wheels while using these incremental encoders. Therefore, custom pc/104 cards were designed to interface to the encoders and control the drive motors.

The boards have four *National Semiconductor* LM629 motion control IC's. The embedded computer interfaces to these IC's via the PC/104 bus. They are dedicated motion control processors that use a quadrature incremental position feedback signal. The optical encoders mounted directly to each drive wheel provide these signals. There are four PWM outputs (each with eight bit resolution), for directly driving an H-bridge motor driver. Each IC may operate in position and velocity mode or velocity only mode. Position and velocity mode will be useful for doing navigation through dead reckoning. Velocity mode will insure that the wheels are all operating at the appropriate speeds even if distance information is not needed [28]. See Figure 3-7.

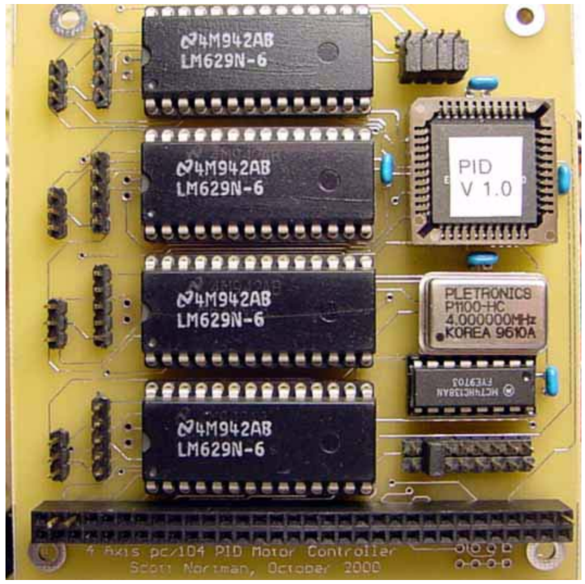

Figure 3-7: Image of LM629 pc/104 card.

3.6 Solid State Motor Drivers

The PWM and PID systems generate the digital control signals for the motors. The control signals are connected to solid state H-Bridge motor drivers. The motor drivers act as amplifiers, allowing the logic level signals control the high current from the batteries. The motor drivers contain four MOSFETs arranged in an H configuration. This allows current to flow in both directions to the motor, controlling the direction of rotation. The switches can turn the current on and of rapidly, controlling the speed of the motors [26]. The details of this control method are discussed in the following chapter. See Figure 3-8 for a block diagram, and Figure 3-9 for an image.

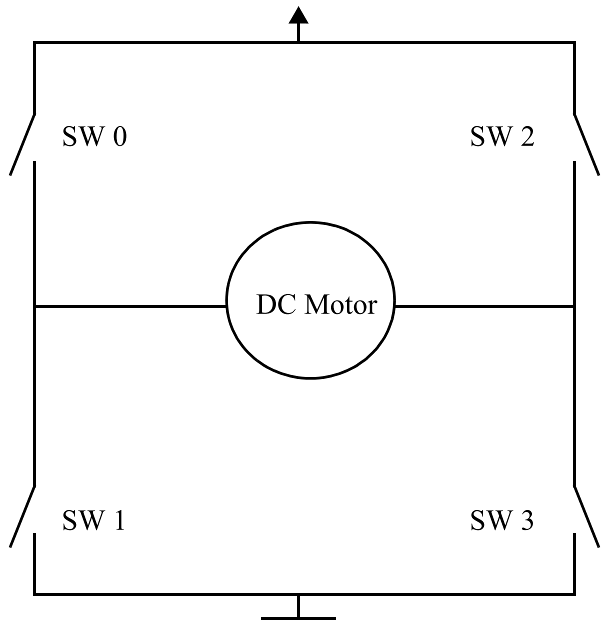

Figure 3-8: Block diagram of the H-bridge motor driver.

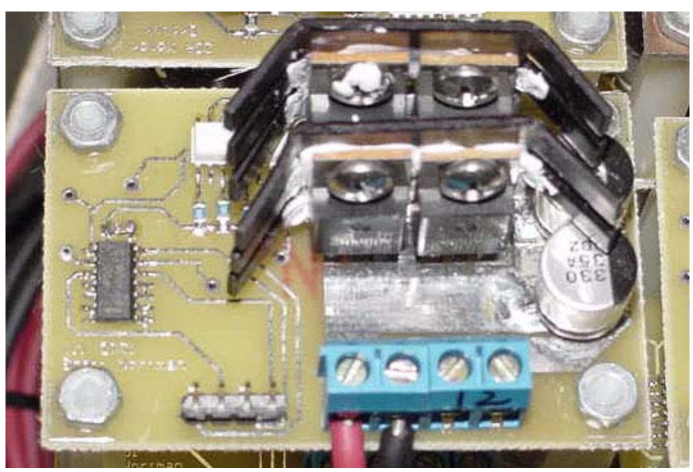

Figure 3-9: Solid state H-bridge motor driver.