Micro-controller Board to support Intelligent Control of Car-like Mobile Robots

Jacky Baltes
j.baltes@auckland.ac.nz

XinKe Lin
xlin001@student.auckland.ac.nz

Centre for Imaging Technology and Robotics
Department of Computer Science
University of Auckland
Private Bag 92019, Auckland, New Zealand

Abstract

This paper describes the design and implementation of an embedded system for the low level control of autonomous mobile robots. The micro controller board provides more accurate speed and direction control, more reliable digital communication, and facilities for additional sensors and actuators. The velocity control is implemented by a one bit D-A converter using pulse width modulation. The data rate of the digital communication is limited to five Bytes/sec., which is sufficient for simple navigation tasks.

1 Introduction

This paper describes the design and implementation of an embedded system for control of remote controlled (RC) cars. The main focus of this research is the design and development of practical systems for navigation of mobile robots in highly dynamic environments. The RC cars are used as cheap autonomous robots. To test our theories, my students and I participate in RoboCup competitions ([2]), an international tournament of autonomous robots playing soccer against each other.

The micro-controller is intended as a low level controller which is supervised by a more powerful single board computer, that is responsible for compute intensive reasoning tasks and the processing of the video input signal.

Therefore, the micro-controller provides the following functionality: motor control, direction control, low level sensor input and digital communication.

Since the DA converters on the toy car are cheap, the speed control of the cars was very coarse. The main problem was that even at the slowest speed setting the vehicles were moving to fast to be controllable by the computer. The use of a pulse width modulation (PWM) motor drive signal allows stable and accurate speed control.

The steering control of the robot is implemented using feed-back control from a standard servo.

The use of a micro-controller allows us to add sensors that are useful in a competitive soccer game such as touch, infrared, and ultrasound.

The cars contain a standard 27 MHz receiver with the corresponding transmitter in the remote controller for the car. We were interested in reusing these transmitters and receivers for agent to agent communication. Interference is a problem with the normal analog transmission, which can severely impact on the performance of our agents in real soccer. Using the micro-controller board, we were able to implement digital communication. The speed of the communication is limited to 5 Bytes/sec. by the design of the original transmitter circuit. However, using a compact encoding, this transmission speed is sufficient for our purpose.

Section 2 describes the processor kernel of the micro controller. The speed and directional control hardware is described in section 3. Another advantage of the board is that it can use digital instead of analog communication. This makes the communication more reliable. The facilities for additional sensors and actuators are described in section 5. The paper concludes with section 7.

2 Car Controller Board

The board uses Motorola's MC68HC11F1 micro controller. The 68HC11 family of micro controllers is very popular. The 68HC11F1 has a separate data and address bus which simplifies the design of circuits with external SRAM and EPROM.

To support more complex control and local navigation algorithms, the car controller board includes 8 KByte EPROM (U1)and 8 KByte SRAM (U2).

The board also contains a Max232 line driver (U10), which implements a standard RS 232 serial interface. This serial interface is used during debugging and also to connect a different computer system to the controller board. Currently, we are working on mounting a standard IBM PC sub-notebook with a USB port camera on the cars. The notebook will be responsible for higher level reasoning and video processing whereas the micro controller is responsible for low level control and navigation.

U11 and U13 are two standard low voltage indicators which are used to implement power on reset as well as manual reset.

Four large diodes (JP2..5) are used to reduce the motor voltage from 9V to 7V. We found in our tests that the car also performs well with 9V motor voltage and so these diodes are not used in the latest version.

3 Speed and Direction Control

The first functionality of the car controller board is to control the steering and the velocity of the toy car. The cars have proportional velocity and steering actuators.

Originally, the velocity was controlled by the generation of an analog voltage between three and seven volts. This mechanism is easy to implement, but provides very poor control at slow speeds. In fact, to overcome the startup inertia of the electric motor, a higher voltage is required than for maintaining the speed. This posed a problem, since the slowest reliable speed setting moved the car at over 1 m/sec, which is so fast that it makes control of the car by the computer very difficult.

A better method for speed control is to use an electronic speed control. Instead of varying the voltage, an electronic speed control varies the duty cycle of a square wave, so called pulse width modulation (P-WM). This PWM method always applies 7V to the motor, but quickly turns the voltage on and off. If the voltage is turned on and off quickly enough (several kHz), it will not induce chattering the mechanical component.

The car controller board uses one parallel output pin PA2 to create the PWM output. This output controls an H-bridge (one half of an L293 (U14A)). The direction of the H-bridge is controlled via pin PA3. The L293 H bridge is rated for 3A maximum sustained current, which is sufficient for our cars. The maximum current may be exceeded however, if the car is stalled.

The directional control of the car is implemented using a standard servo. The servo motor is controlled via a separate H-bridge (U14B) using pins PA0 and PA1. The resistance feedback of the servo is feed to pin PE1, an on chip A-D converter.

4 Digital Communication

The RC cars use a standard two proportional channel control which is modulated with a 27 MHz carrier wave.

The original transmitter used two proportional channels. The receiver outputs an analog voltage. The analog voltage is converted into a digital input signal using a Schmitt trigger (U16).

The second channel is used to control the servo in the original receiver circuit. A digital signal for the second channel is generated by connecting the resistance feedback to a fixed resistor. In this way, if the transmitter generates a full left or full right signal, the receiver generates positive or negative 7V.

The digital communication uses both channels. The clock is transmitted using the directional data channel, whereas the data is transmitted on the velocity channel.

The maximum transfer rate is limited by the transmission rate of the transmitter, which is 20 ms. The maximum bit rate is therefore limited to 50 bits per second. Using a standard serial protocol with one start bit and one stop bit leads to a raw maximum transfer rate of 5 Bytes/sec.

The speed of communication is further limited through the fact that the transmitter is not synchronized to the PC controlling it. The maximum package size is limited by the clock skew between the PC and the controller.

The digital communication protocol that we implemented use the second channel to transmit the clock. The problem of clock skew can thus be alleviated. Nevertheless, the current message size is limited to four bits to limit the impact of interference.

5 Sensors

The car controller board was designed with future expansion in mind. In particular, it includes four free bits that can be individually programmed as input or output. Furthermore, it has four free analog input ports. These ports were intended for future addition of simple sensors, such as touch sensors, ultrasound, or infrared. Currently, the cars were modified and simply touch sensors for the front and the rear of the cars were added. The board can also be used to control simple actuators. The students implemented turn signals and horns using these spare outputs.

In the future, we would like to use sensors to support more accurate navigation such as shaft encoders and stall detectors.

6 Firmware

This section describes the implementation of the firmware of the car controller board. The control of a vehicle in a real world environment imposes a number of hard and soft real time constraints. The design of such a system as a monolithic application is difficult and error prone. Instead our approach is based on the notion of tasks. Each task has a priority (either high or low) associated with it. Figure 1 shows the architecture of the firmware.

The firmware uses a small task switching kernel for low priority tasks. The kernel driven by a slow timer interrupt (50 Hz). It also uses a high priority timer interrupt to control high priority processes. Data is passed between different tasks using a global parameter blocks rather than through the use of queues. This greatly simplifies the design, development, and debugging of the firmware. It has the disadvantage that data may need to be thrown away. For example, if the digital communication receives a new speed setting, it will store it in the speed setting block. If the car controller receives another speed setting before the high priority speed interrupt has had a chance to set the new speed setting, it will overwrite the previous speed setting. The motivation for this design is that the car controller is intended as a slave processor which is controlled by a higher level processor (Master processor). Therefore, the goal is to reduce the latency of commands being send by the higher level processor and to execute commands as quickly as possible. However, each parameter block has a lock bit associated with it to avoid corruption of data blocks of more than one byte.

The task switching kernel is responsible for touch sensors, sound generation, lights, and local navigation. The touch sensor is implemented as a low priority task, since the car moves slow. Given a faster or expensive car, it may be necessary to convert this into a high priority interrupt task. The sound generator and lights control the horn and turning signals. The local navigation task executes a small set of macros that are designed to move a car around an obstacle. For example, if the car touches an obstacle in the front, then back away by 10 cm and turn slightly to the left. The navigation system is also responsible for turning the lights and horns on.

The high priority timer interrupt is generated every 1 ms. The high priority timer interrupt is responsible for speed control, direction control, A-D conversion, and digital communication. The speed control uses an internal counter register to generate the P-WM wave form. The high speed timer interrupt only has to reload the register values. The directional control uses a simple controller to control the servo. The analog is converted into a digital value. If the current resistance is not equal to the desired resistance, the directional motor is turned on. The direction of the motor is determined by the sign of the difference. The high priority interrupt is also responsible for the A-D conversion.

7 Conclusion

The described controller was developed during Jan. and Feb. 1999 and was used in the paper "415.703 Intelligent Active Vision," which is taught at the CITR. The controller has proven to be reliable and

Control of the cars, especially at low speeds, is greatly improved. Steering control is only limited by the resolution of the built in A-D converter (8 bits) and the jitter in the servo motors themselves.

The biggest problem was the implementation of the digital communication. Since the transmitter and the PC are not synchronized, the communication speed has to be reduced to about 5 Bytes/sec. Also, the clock skew limits the maximum packet length to about 4 Bytes. Nevertheless, even this limited amount of communication has been useful.

The students used the controller board to compete in the Merc Mouse Competition [1]. The Merc Mouse Competition is similar to the popular micro-mouse competition. The task is to navigate through a maze as quickly as possible. However, it is difficult to install all necessary sensors for a large robot (40cm length)

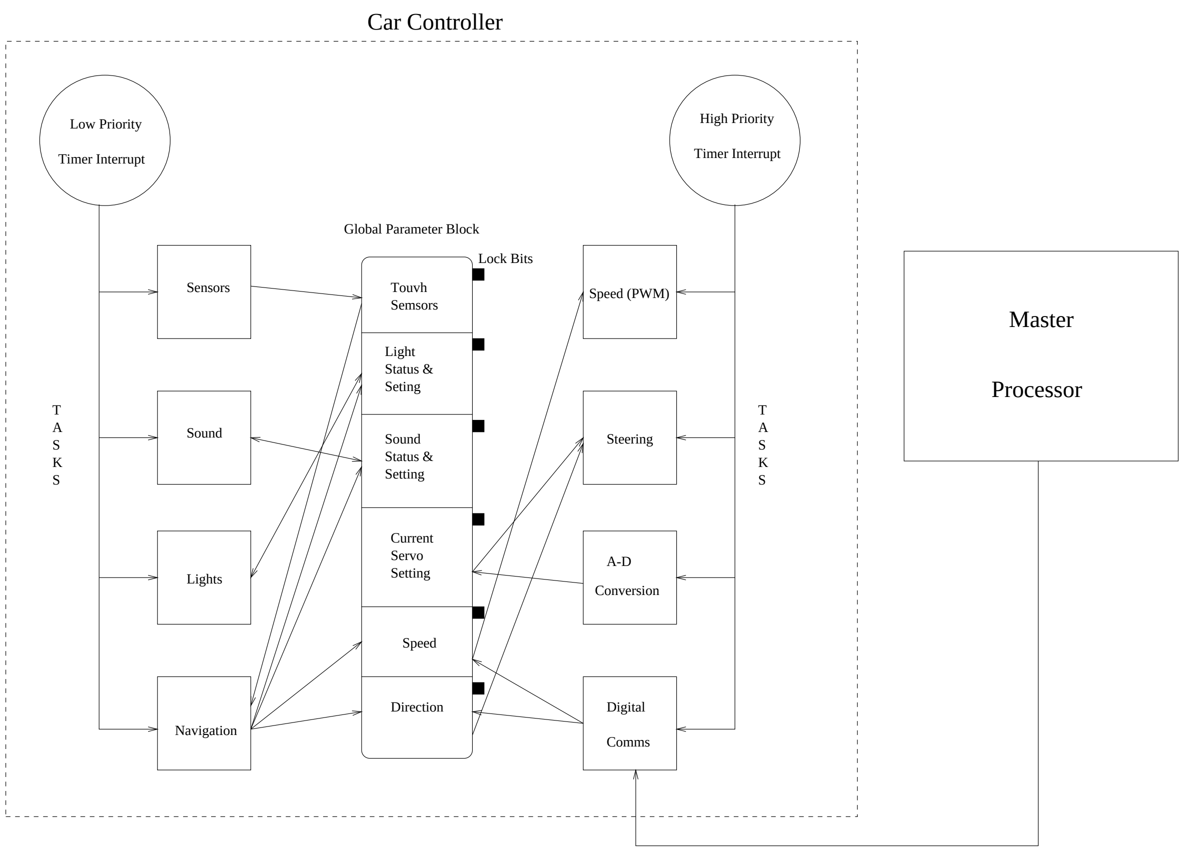


Figure 1: Architecture of the Firmware

to detect all possible stall conditions. For example, by cutting a corner, a car may get stuck against an obstacle without triggering the front and rear touch sensors. Therefore, the Merc mouse competition uses a global vision system installed on the ceiling to provide information about the progress of the robot towards the goal.

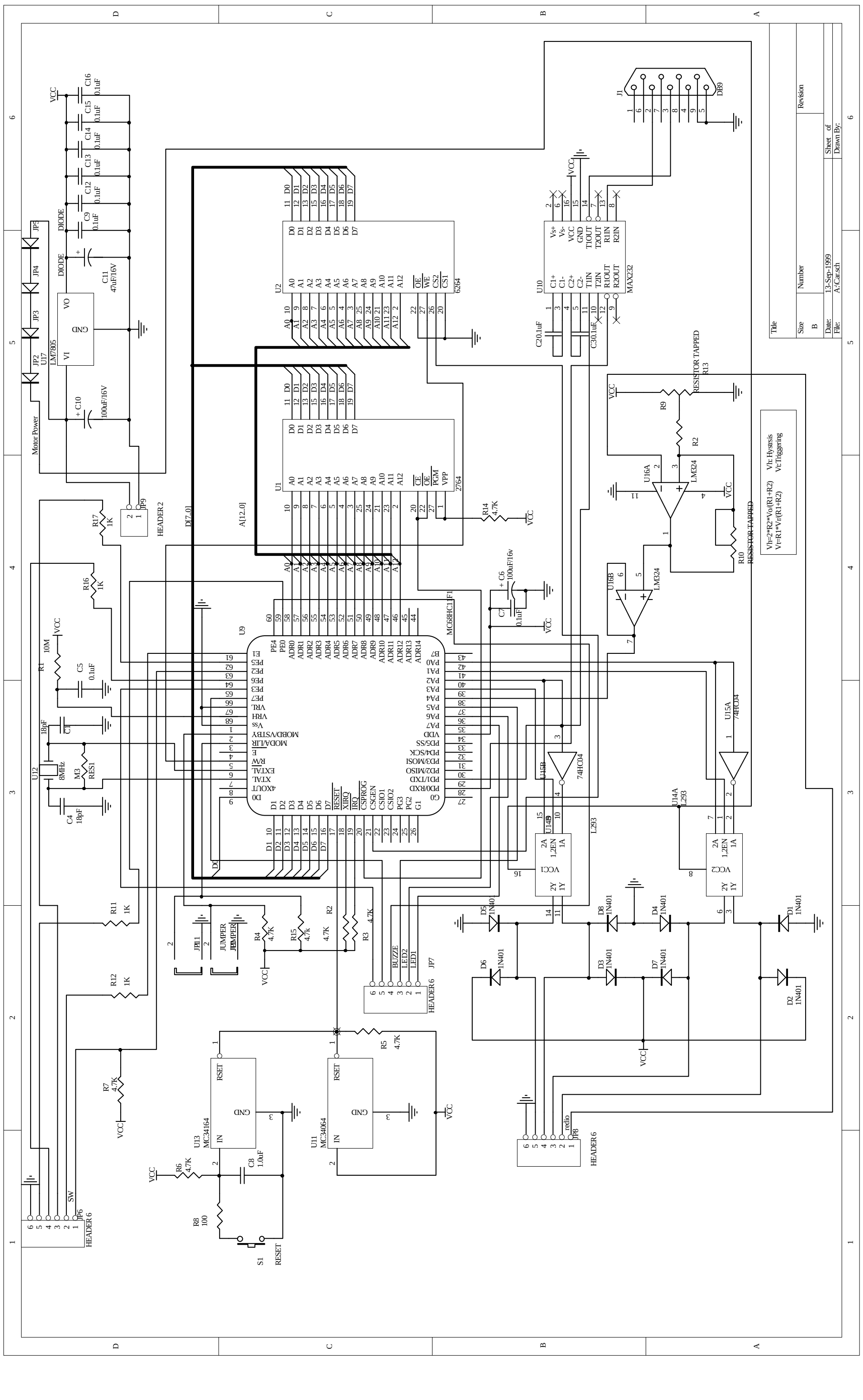
The digital communication was reliable enough to transmit information from the video server to the car to navigate the car through a maze.

In the future, we intend to mount a single board PC on the cars. Local vision information will be provided by a USB port camera. The single board PC is used to interpret the local vision information, for example to detect walls, balls, or other objects. Using this local vision information, the PC will generate high level goals (e.g., approach an obstacle) and plans (e.g., back out of the corner). The PC uses the serial interface to communicate with the micro controller which is responsible for low level navigation (e.g., detection of obstacles straight ahead, turn 10 degrees to the left).

References

[1] Jacky Baltes. The merc-mouse competition. http://www.citr.auckland.ac.nz/ jacky, April 1999.

[2] Jacky Baltes, Nicholas Hildreth, and Yuming Lin. The all botz robocup team. In *Proceedings of the I-JCAI Workshop on RoboCup*, Stockholm, Sweden, July 1999.


Title
Size
B
Number
Revision
Date: 13-Sep-1999
File: A:\Carsch
Sheet of
Drawn By:
MC68HC11F1
U9
U1 2764
U2 6264
U10 MAX232
J1 DB9
U17 LM7805
U16A LM324
U16B LM324
U15A 74HC04
U15B 74HC04
U14A L293
U14B L293
U11 MC34064
U13 MC34164
S1 RESET
HEADER 6
HEADER 2
Motor Power
Vh=2*R2*Vo/(R1+R2)
Vt=R1*Vr/(R1+R2)
Vh: Hystersis
Vt: Triggering
RESISTOR TAPPED
BUZZE
LED2
LED1
JUMPER
radio
SW
8MHz