CpE 180
SPRING 2001

CAN BUS DIAGNOSTIC TRANSLATOR PROPOSAL

TEAM NUMBER 19

BUDGET TOTAL
$180

NAMES OF GROUP MEMBERS
KENNETH HECK
JOHN MURPHY

FACULTY MONITOR
PROFESSOR WILS L. COOLEY

SPONSORS
PROFESSOR ROY S. NUTTER
PROFESSOR G. MICHAEL PALMER

04/23/2001

Contact Person:
Kenneth A. Heck
Route 2 Box 182
Fairmont, WV 26554
(304) 363-6824; (304) 363-2814
kheck2@wvu.edu

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

Throughout the 1970's and 80's, the electrical systems in automobiles dramatically increased in complexity and weight with the continuing addition of electrical devices such as sound systems and on-board computers, and with the increasing use of electrical systems for the control of systems that were previously purely mechanical, such as braking. These devices and systems required correspondingly sophisticated control and communications systems, especially as their role in the operation of the car became more central. These developments meant difficulty in the design and maintenance of new automobiles, and often meant that people could no longer repair their own cars. The devices themselves became heavy and complicated, and the opportunities for malfunction increased.

In an effort to simplify the design and maintenance of new automobiles, the Controller Area Network (CAN) standard was invented at Robert Bosch GmbH (Reutlingen, Germany) in the late 1980's, and later the CAN 2.0 standard in September 1991.[1] The problems they faced included the need for safety, the high-noise environment of a running automobile, and the need for flexibility and speed of high-fidelity communication among electrical devices in a car. The solution was a high-speed bus (Referred to as a CAN bus), the signals on which traveled on a pair of polarity-switching lines, and which was capable of extremely reliable performance.[2]

Adoption of the CAN standard by automobile manufacturers came eventually, but the technology found enthusiastic support in other and varied settings, including the fields of industrial automation and medical equipment. Each field that adopted the CAN standard did so with its own small changes and refinements, to the degree that today some implementations have little in common.[2]

The Formula Lightning team at West Virginia University is one such user of the CAN standard. Led by Dr. Roy Nutter, the Formula Lightning team has devoted itself to the design and construction of a fully electric racecar. This car uses a CAN bus for the control and operation of its electrical systems. In order to ensure the correct and optimal operation of these systems, a means of access the bus must be devised. The bus must be accessible to them under both workshop and field conditions in order to diagnose and solve the various electrical problems that arise.

SECTION 2. DESIGN OBJECTIVES

2.1 Design Goals and Constraints

The Formula Lightning team requires a device to perform the task of communication and control on their implementation of the CAN bus. This device must translate CAN signals to a standard signal that can be read by a technician with a laptop computer running a software package developed by Andy Pertl. The device must enable the technician to read signals on the bus, whether all the signals or only a selected set, and to write signals to the bus.

While this alone is sufficient for workshop use, the nature of the application requires field testing – literally, while racing the car around a track – and so further constraints are placed on the operation of the device. Due to the economy of space in the Formula Lightning car, the device must operate without a technician's computer connected while the car is in motion. The device must itself, therefore, be small enough to not encumber the driver of the car or otherwise interfere with the car's normal operations.

In order to be of use with this particular implementation of the CAN bus, the device must communicate according to the SAE standard described in J1939. This will enable it to correctly read a signal from a CAN bus line. The signals it must read and write to the bus may be either or both CAN 2.0A and CAN 2.0B, which are similar packet structures differing mainly in length. In order to communicate with the laptop computer, the device will communicate to the computer's COM port using the RS-232 standard, using the protocols accessed by Andy Pertl’s software package. A microcontroller capable of Universal Asynchronous Receive/Transmit (UART) combined with a single-chip RS-232 device will be suitable for this function.

The act of writing a signal to the bus requires that the device read the signal from the laptop, and package it correctly for CAN 2.0A and CAN 2.0B. This will require the microcontroller to assemble a full CAN packet (which is larger than the packets transmitted by the laptop) and construct the correct header depending on whether the packet is 2.0A or B. Once correctly packaged, the device will write the signal to the bus according to the arbitration methods defined in the CAN standard.

The signal that the device writes may also be stored on the device in such a way that it can be "played back" onto the bus at a pre-determined time, or multiple times at predetermined intervals. This will enable the device to be used without an attached computer for a short period of time. In order to perform this task, there will be a series of pre-defined instructions that the device must recognize, accept, and process to regulate its own behavior and state. These signals will be defined in such a way that the device will be able to quickly distinguish them from signals that must be written to the CAN bus.

In order to read from the bus, the device will take a signal in its entirety from the bus line, and determine whether the signal is wanted by the user. If the signal is wanted, the device will either divide it up to send to the attached computer via RS-232, or if a computer is not attached, store it for later playback.

In playback mode, the device will communicate the stored signals in the order that they were received to the attached computer in such a way as to simulate the CAN bus's operation during the recorded time. This mode will be easily accessible to the user of the software, and removable storage media will allow multiple playbacks of various test runs.

2.2 Design Specifications

Input Impedance:
CAN-side: .... 120Ω +/- 12Ω

Device Power Requirements:
Power Sources: .... External: 12V w/ shielded cable
Current Draw: .... <200mA

Communications:
CAN-side: .... Max 1Mbit/sec
PC-side: .... Max 115.2Kbit/sec

Communications Protocols:
CAN-side .... (1) CAN 1.x
(2) CAN 2.0A
(3) CAN 2.0B (Read Only)
PC-side RS-232

Connections:
CAN-side DB-9
CAN-side connector harness .... (1) DB-9 to 5-pole
(2) DB-9 to 6-pole
(3) DB-9 to 9-pole
PC-side DB-9
PC-side connector harness .... (1) DB-9 to DB-9
(2) DB-9 to DB-25
Power .... 2.1mm barrel

Operational Environment:
Electrical Isolation .... Max 24V applied to case
Shock (Force) .... Max 4g
Pressure .... Max 60psi
Water Resistance .... 1ft submersible (or equivalent)
Temperature:…………………..……………Between 0, 100 degrees Fahrenheit

Dimensions (excluding external cabling):
Case .... 2.00” H x 5.00” W x 6.00” D
(51mm H x 127mm W x 153mm D)
Weight .... 3.0 lbs (1.37 kg)

PC Requirements:
Must run CAN software written by Andy Pertl
Programming Languages:
PC:…………………………………………C or C++
PIC:…………………………………………PIC C, Assembly where needed

2.3 Deliverables

The device, when finished, will consist of a box as described above -- no larger than 2"x4"x5", weighing not more than 3.00 pounds, with four openings. The first opening will allow a cable to connect the device to the CAN bus via DB-9. The second opening will be a DB-9 type socket for a cable connecting to a computer. This second opening will be pluggable to prevent entry of dirt, oil or water. The third opening will allow a power cord. The last opening will permit a flash memory card. This opening will also be sealable.

The power cord and CAN bus cable will be provided with the device. The user must provide the cable connecting to the DB-9 socket. One flash memory card will be provided with the system. Additional cards of the same type may be used with this device, but purchase, storage, and care of these cards is solely the responsibility of the user.

A manual will also be delivered containing detailed instructions of use and maintenance for the device, as well as a complete listing of all software code and circuit diagrams, with a complete listing of parts used. Repair directions will be given for parts more easily replaced, such as fuses.

A sketch of the proposed product is attached in the appendices (Figure 5).

2.4 Validation

In order to determine that the device works correctly, the following tests will be performed:

The delivered device will be connected to a laptop computer and a working CAN bus. This same bus will have on it a similar device (of outside manufacture) for testing purposes. First, the system will be run so that the bus will carry signals from multiple sources. The device being tested will identify all of these signals and relay them. The signals relayed by the delivered device will be compared to those relayed by the outside device to ensure that all signals have been received, and no extraneous signals reported. This comparison will be done using two PCs running identical software, connected to the same bus, one with the delivered device and one with the outside device.

The delivered device will then be instructed to send certain signals over the shared bus. The outside device's output will be consulted to verify that these signals have been sent correctly. The delivered device will be given instructions by its PC to store all signals from a given source on the bus (Identified using a CAN numeric identifier) and then attached to a CAN bus without the computer attached. The system will be run so that the bus carries signals from this source for two minutes, then shut down, and the delivered device attached to a computer to relay the stored data.

Lastly, the device will be used for a week at the test track under normal-use conditions, which may include accidental dropping or wetting. After this week, the device will be put through the above tests once more to ensure that no harm has come to it.

SECTION 3. SYSTEM DESCRIPTION

3.1 System Description

The device will be a single-board electronic device with a 12-volt DC power supply (external, drawing from automobile battery). It will contain a 40 MHz oscillator for a clock, and four microchips: A PIC microcontroller (likely the 16F877, referred to as "additional PIC" on the budget), Microchip's CAN bus controller MCP2510, a Max232 IC for RS-232 and an Intel 82527 for the CAN voltage conversion. Circuits for power conversion will be present, and a fuse will protect the circuit from power spikes. There will also be a socket for connecting a flash memory card.

The PIC chips will handle the tasks of interfacing with the CAN bus, interfacing with the computer (via RS-232), storing information to the flash memory, receiving and executing instructions from the computer, and both determining then communicating error states. Microchip's CAN controller auto-detects certain errors and can "mask out" packets from certain sources. This controller will handle arbitration and communications on the CAN bus. Once the controller has received a full packet, it will pass it to the PIC, which can be programmed to either store it on the flash memory for later retrieval, or (by default) to break the packet into smaller portions and send them serially via RS-232. The device will activate immediately upon being plugged into a power source, and will have a sleep mode to conserve power when it is idle.

The PC-side connection will be a female DB-9 connector, electrically isolated from the CAN bus. The CAN-side connection will be a male DB-9 connector. Both will be clearly marked to prevent confusion. The flash memory card will be connected via its proprietary connector. The power cable, which will be heavily insulated, will be attached using a 2.1mm barrel connector and clearly marked as “Power.”

The program on the PIC will be done in PIC C, with inline PIC assembly for the speed-critical sections. Microchip's MPLAB software and header files will be used for programming the PIC. On the PC, Dynamically Linked Libraries (DLLs) provided by Andy Pertl will be used, most likely in conjunction with the Visual C++ and Visual Basic Studios, both designed for programming in the Windows 98 and Windows NT environments. All source code will be delivered in hard copy to Dr. Nutter with the manuals for the device, with compilation instructions.

3.2 Block Diagram

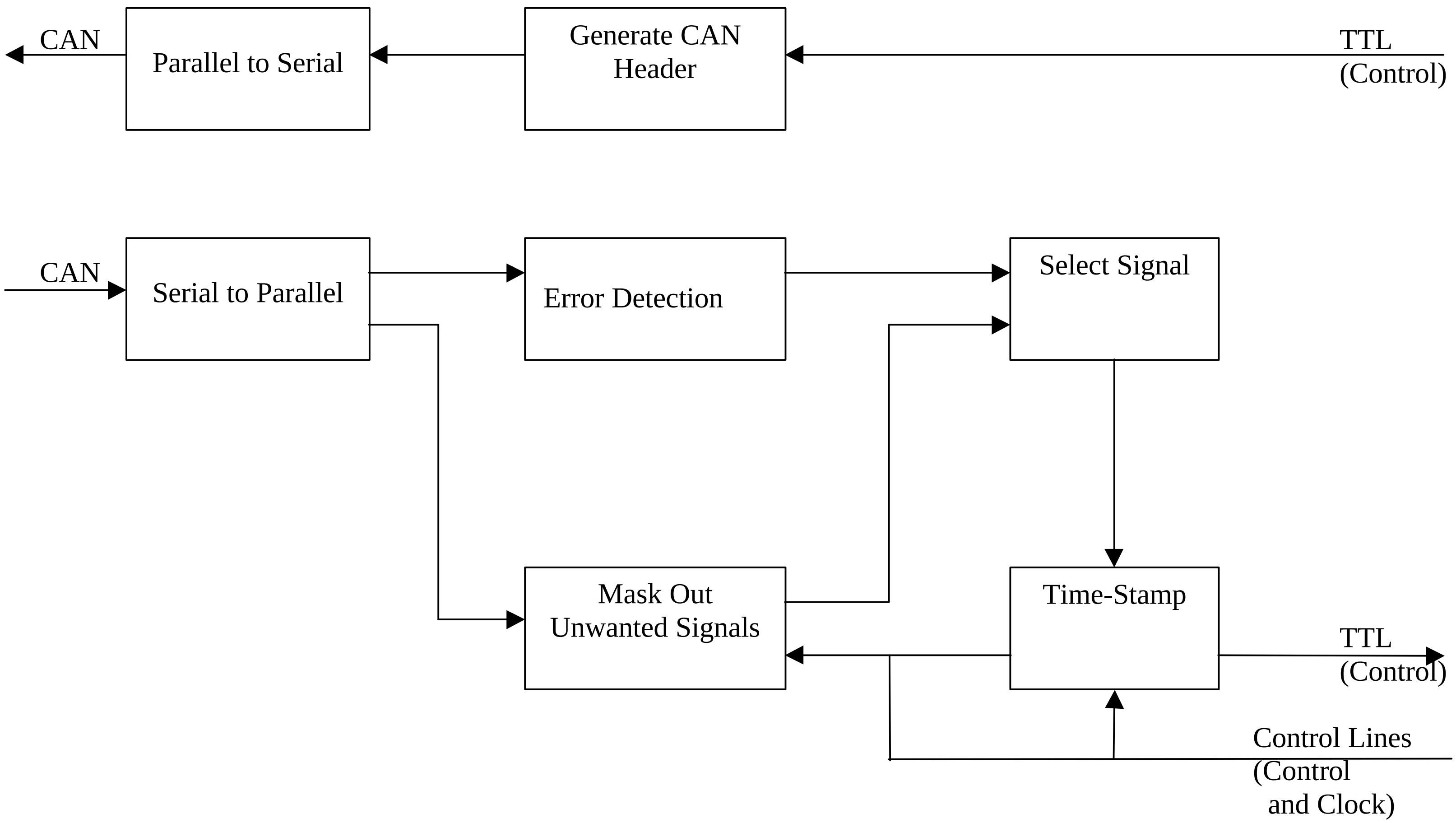
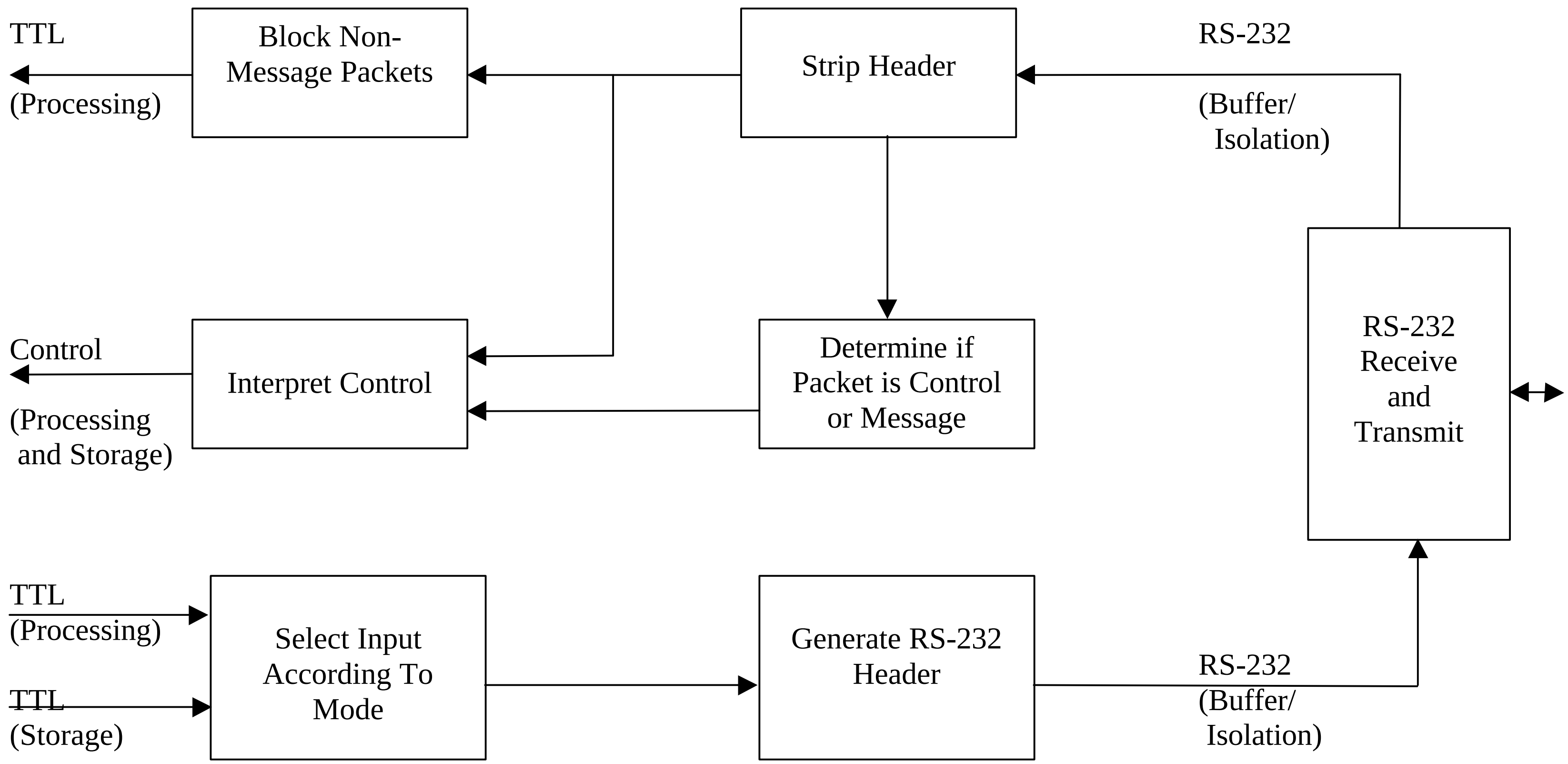
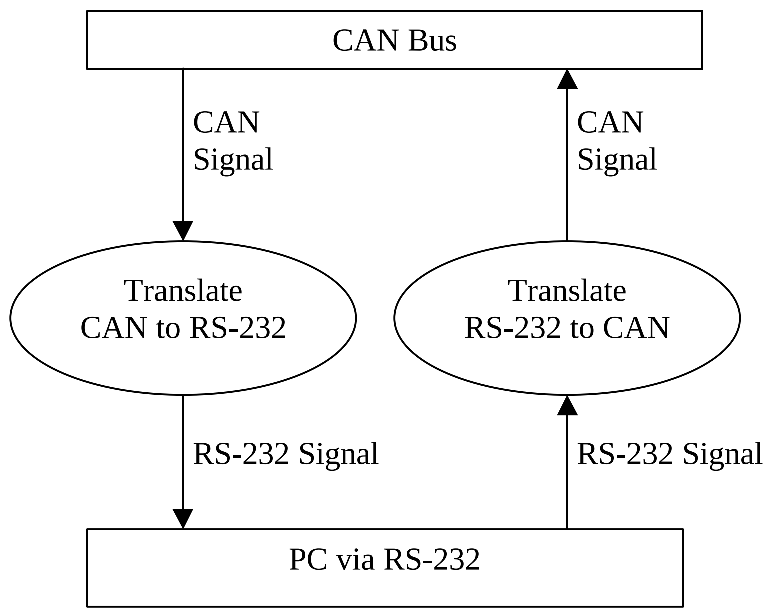


Figure 1: CAN Bus Diagnostic Translator Block Diagram

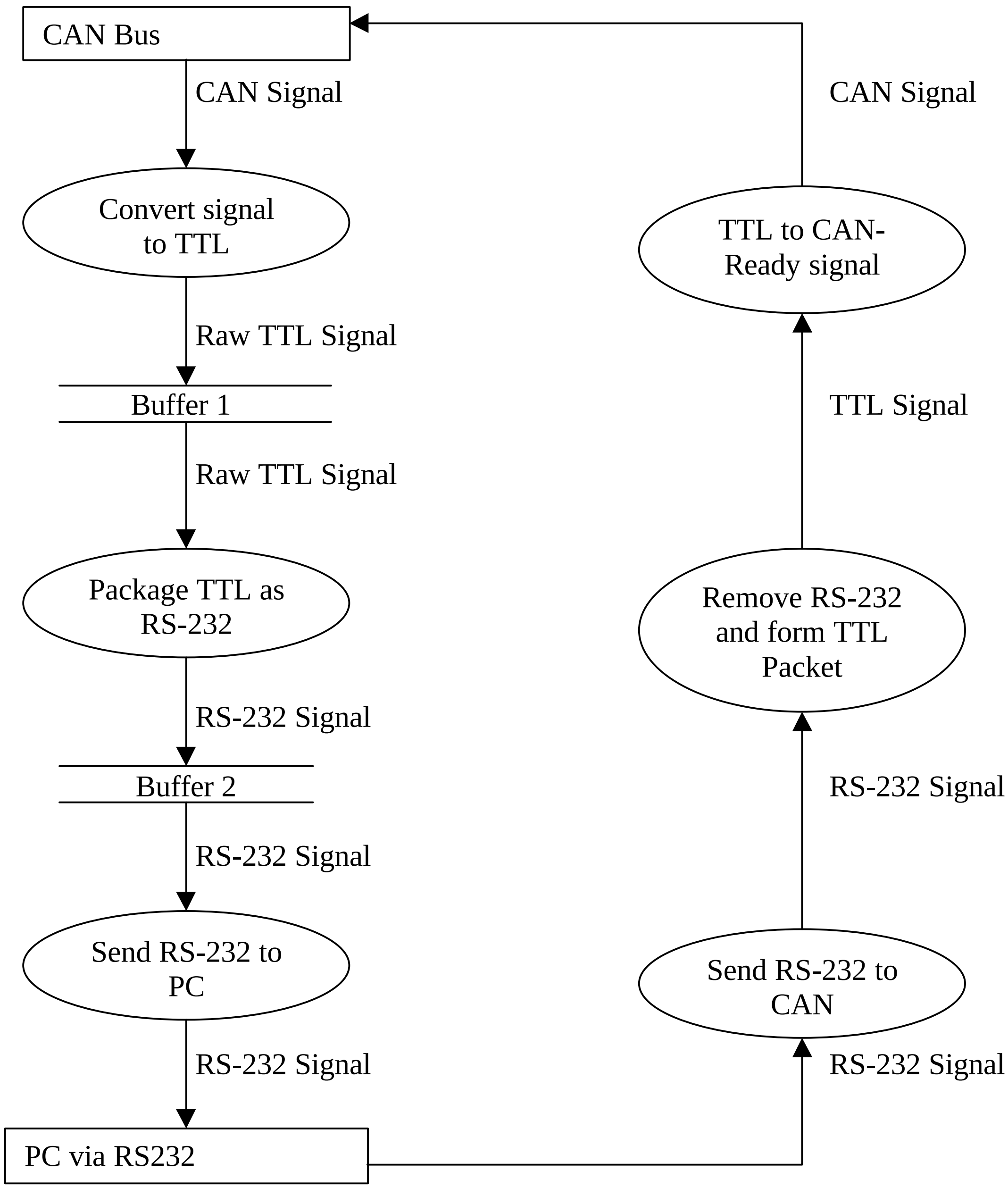


3.3 Data Flow Diagrams



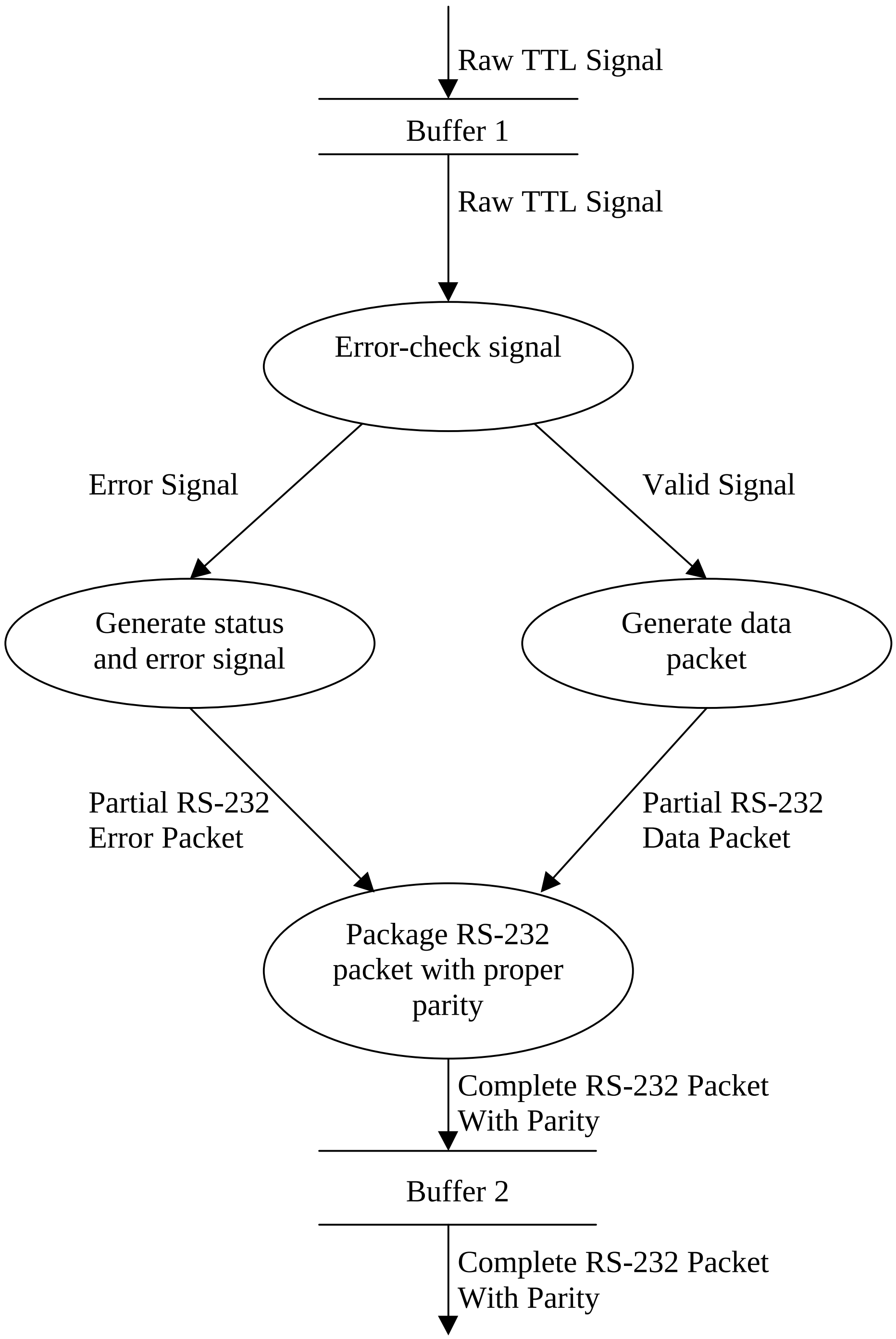
Level 0 DFD

Figure 2: Level 0 Data Flow Diagram



Level 1 DFD

Figure 3: Level 1 Data Flow Diagram



Level 2 DFD

Figure 4: Level 2 Data Flow Diagram

SECTION 4. ORGANIZATION AND PLANNING

4.1 Scheduling

While the following Gantt chart assumes that all work will be done in the Fall semester, likely many of the tasks will be performed during the previous summer. Many of the tasks to be performed are self-evident – such as the purchase of parts. There are a few things on the Gantt chart that bear explanations.

The chart refers to “PIC #2” in some tasks. This PIC is the “additional PIC” in the budget list, and takes much of the responsibility for the left-hand side of the block diagram, notably clock and control functions.

As noted in the Deliverables section, only one manual will be produced, as per Dr. Nutter’s instruction. This manual will combine the documents normally referred to as the “user’s manual” and “service manual.” This combination is for his convenience and for a general reduction of redundancy in documentation. They are listed separately because of the nature of the work, even though they will at the end be combined into a single deliverable document.

Lastly, the number/number notation is shorthand for the person responsible, in the order Heck/Murphy. Thus, 10/5 in a block represents Heck spending 10 hours and Murphy spending 5. A /3 in a block represents Murphy alone spending 3 hours.

4.2 Division of Responsibility

Task	Effort		Week														
	KH	JM	08/20	08/27	09/03	09/10	09/17	09/24	10/01	10/08	10/15	10/22	10/29	11/05	11/12	11/19	11/26
1) Design Circuit	5	5	5/5														
1) Buy Parts	1	1		1/1													
2) Learn to use and wire CANBus PIC	20	5		5/5	10/	5/											
3) Program the CANBus PIC	20	0				5/	15/										
4) Learn to use and wire PIC#2	20	5						10/5	10/								
5) Program PIC#2	0	20							/10	/10							
6) Learn to use and wire memory card	20	10								10/5	10/5						
7) Learn to use Andy Pertl's software	10	10		10/10													
8) Make changes to AP software	0	40			/10	/10	/10	/10									
9) Design power supply	5	0	5/														
10) Learn to use and wire the serial port	0	5							/5								
11) Learn to etch boards	5	5										5/5					
12) Learn to build/etch prototype board	10	10										5/5	5/5				
13) Build case to meet specifications	3	3											3/3				
14) Test project to ensure specifications	5	5												5/5			
15) Service Manual	10	10		1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1					
16) User's Manual	10	10		1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1					
17) Prepare for Design Fair	10	10													10/10		
TOTAL	154	154	10/5	18/18	12/12	12/12	17/12	12/17	12/17	12/17	12/12	12/12	8/8	5/5	10/10		

SECTION 5. BUDGET

5.1 Budget

	Project Cost	Proposal Cost
1. CANBUS Controller – MCP2510	$7	$7
2. Additional PIC for other Functions	$10	$10
3. PICStart Plus Programmer	$150	$0*
4. MAX232A	$5	$0**
5. Intel 82527	$9	$9
6. DB9 connector (Receptacle)	$3	$3
7. DB9 connector (Plug)	$3	$3
8. Circuit Board Etch Kit	$30	$30
9. Capacitors, Resistors, Inductors	$5	$5
10. Case (Aluminum)	$15	$15
11. Power Connector (Receptacle)	$1	$1
12. Power Connector (Plug)	$4	$4
13. Oscillator	$3	$3
14. IC Sockets	$1	$0**
15. Flash Memory (32MB)	$40	$40
16. Flash Socket	$5	$5
17. DC-DC Converter	$9	$9
18. Resettable Fuse	$1	$1
19. Power Switch	$5	$5
20. Any other discrete components	$5	$5
21. Miscellaneous (solder, etc.)	$5	$5
22. Shipping/Handling Charges	$20	$20
TOTAL	$336	$180

Requested from Sponsors=$80

Requested from CSEE Department=$100

Sponsor Signature Date

Chair Signature Date

Monitor Signature Date

5.2 Budget Justifications

1. CANBUS Controller – MCP2510	Digi-Key # MCP2510-E/P-ND
2. Additional PIC for other Functions	Digi-Key # PIC16F877-20/P-ND
3. PICStart Plus Programmer*	Digi-Key # UP003001
4. MAX232A**	Digi-Key # MAX232ACPE-ND
5. Intel 82527	Pioneer Std # AN82527F8
6. DB9 connector (Receptacle)	Newark # H3R09RA29B
7. DB9 connector (Plug)	Newark # H3M09RA29B
8. Circuit Board Etch Kit	Newark # 00Z1482
10. Case (Aluminum)	Newark # 95F939
11. Power Connector (Receptacle)	Newark # 84N1192
12. Power Connector (Plug)	Newark # 84N1158
13. Oscillator	Newark # 95F8736
15. Flash Memory (32MB)***	Buy.com # SDCFB-32-455
16. Flash Socket***	Newark # 95B8852
17. DC-DC Converter	Newark # 95B9538
18. Resettable Fuse	Newark # 85F256
19. Power Switch	Newark # 46F3056

* Provided by Kenneth A. Heck

****** Donated by Kenneth A. Heck

******* Compact Flash used for this budget, though a different flash memory may ultimately be utilized.

SECTION 6. QUALIFICATIONS

6.1 Kenneth Alan Heck, MS

Kenneth Alan Heck, MS
Route 2 Box 182
Fairmont, WV 26554

Home Phone : (304) 363-6824
E-Mail: kheck2@wvu.edu

Education :

01/98-Present:

University :	West Virginia University, Morgantown, WV
Coursework :	Pursuing BS in Computer and Electrical Engineering

08/96-12/96 :

College :	Santa Fe Community College, Gainesville, FL : EMT - B, December 1996
Registration :	National Registry of Emergency Medical Technicians - #B1103946
License :	West Virginia - #B-036323

08/92-02/96 :

University :	University of Florida, Gainesville, FL : MS, December 1995
Thesis :	"General Synthetic Methods for Alpha-hydroxy Ketones"
Research Director :	Dr. Alan R. Katritzky, Kenan Professor, Center for Heterocyclic Compounds
Graduate Coursework :	1 sem. each : General Organic, Organic Synthesis, Organic Mechanism, Organic Spectroscopy, Inorganic, Organometallics, Quantum Theory
Equipment Skills :	300MHz NMR's : Varian Gemini-300, VXR 300, GE QE-300; HPLC
Other Duties :	Katritzky group lab steward, miscellaneous equipment repair

08/88-05/92 :

University :	West Virginia University, Morgantown, WV : BA, Chemistry December 1992
Coursework :	Mostly BS and Honors courses, organic courses with microscale techniques
Special Topics Course :	X-Ray diffraction
Equipment Skills :	IR : Perkin-Elmer; PC-based systems
Honors/Scholarships :	(i) WVU Presidential Scholarship (ii) John A. Moore (Chemistry) Scholarship (iii) WVU Honors Program

Experience :

01/00-Present :

Business :	Heck Solutions
Title :	Owner
Functions:	Software Development specializing in Access 97 Custom-built PC's/Upgrades PC Diagnosis/Repair

08/97-12/99 :

Employer :	West Virginia University, Financial Aid Office, Morgantown, WV 26506-6004
Title :	Information Systems Technician
Supervisor :	Tresa Weimer, Supervisor : (304) 293-8571 ; 1-800-344-WVUl
Duties :	System Administrator of Local Area Network running Novell Netware 4.11 Training employees to use software : Netware 4.11, Win 95, Win 98, WP6.1, FoxPro 2.6, BANNER 2.15, MS Office Pro 97 Maintenance and Repair: Various PC's and Printers Developing need-specific software for varying uses in the office : Access 97 , FoxPro 2.6 Supervision of Work Study employees

06/97-Present :

Employer :	Monongalia General Hospital, 1200 JD Anderson Drive, Morgantown, WV 26505
Title :	Part-Time Monitor Tech
Supervisor :	Glenda Broad: (304) 598-1506
Duties :	Watching 5 monitors capable of 8 strips each, paging nurses for dysrhythmias Recording and analyzing rhythm strips once per shift Tracking patient room assignments, notifying necessary personnel for admits/discharges Maintaining patient information board for physicians, nurses, administrative personnel

06/96-02/97 :

Employer :	Farchan Laboratories, 4906 NW 53rd Street, Gainesville, FL 32653
Title :	Bench Chemist - Research and Development
Supervisor :	Dr. Radi Awartani : (352) 374-6825
Duties :	Running small (50mL) to large (22L) scale reactions as directed Supervising technicians in running reactions Quality control analysis by GC, HPLC, IR, Karl-Fischer water analysis, titration, melting point and refractive index Lab equipment maintenance

01/93-02/96 :

Employer :	University of Florida, Department of Chemistry, Gainesville, FL 32611
Title :	Graduate Research/Teaching Assistant
Supervisor :	Prof. Merle A. Battiste : (352) 392-0552
Duties :	3 sem. General Chemistry, 2 sem. Organic Chemistry Lab

08/88-07/92 :

Employer :	West Virginia University, Financial Aid Office, Morgantown, WV 26506-6004
Title :	Clerical Assistant
Supervisor :	Brenda Thompson, Director : (304) 293-5242 ; 1-800-344-WVUl
Duties :	Clerical work : Filing, typing, answering phones, stocking, errands Developing need-specific software for varying uses in the office : FoxPro 2.0 LAN Training employees to use software : Netware 2.10, WP5.1, FoxPro 2.0, DOS PC Maintenance

03/91-07/92 :

Employer :	First National Bank of Morgantown, 201 High Street, Morgantown, WV 26505 (now Huntington Banks, WV)
Title :	Part-Time Computer Operator
Supervisor :	Scot Epling, Assistant Vice President : (304) 367-2452 ; 1-800-377-BANK
Duties :	Unsupervised after hours final sorting of personal checks using a Honeywell DPS 6 Developing need-specific software : dBase IV 1.1 and QuickBasic Software installation : WP5.1, Lotus, PCTools Diagnosing and correcting PC hardware problems

Publication :

A. R. Katritzky, K. A. Heck, J. Li, A. Wells, C. Garot, "1-(1-Alkenyl)benzotriazoles: Novel Equivalents for the Synthesis of α-Hydroxy Ketones", *Synth. Commun.*, 26(14), 2657-2670. (1996)

6.2 John Murphy

John Murphy
910 Montrose Ave
Morgantown, WV
26505
(304)292-8870
murphyj@csee.wvu.edu

Resume

Objectives:

To obtain a graduate degree in electrical engineering in the particular field of robotics.
To gain experience designing, building, controlling, and maintaining robotic systems.

Job Experience:

Intern, NASA IV&V Summer, 1996
Fairmont, WV (Funded by George Washington University)

Involved with SORT (Software Optimization and Reuse Technology) project, a software engineering effort as related in particular to Marshall Space Flight Center's flight furnace project.
Oral demonstrations of progress given both to peers and superiors at the end of the internship

Math tutor, West Virgina University Math Department Aug 2000 - present
Morgantown, WV

Education:
West Virginia University 1997-present
Morgantown, WV
(Expected graduation date: Dec 2001)
Degrees to be received:
BSCpE, BSEE, emphasis in control systems

Morgantown High School 1994-1997
Morgantown, WV

Skills:

Programming (Windows, MS-DOS and UNIX environments):
C, Java, Intel x86 assembly, PIC assembly, PIC C, Ada, Matlab, Perl, Lisp
Spoken and Written Japanese
Use of fuzzy logic and genetic algorithm techniques, particularly in control applications
Experience designing with and programming for PIC chips

SECTION 7. REFERENCES

7.1 Notated References

1) CAN Specification, Version 2.0, Robert Bosch GmbH, Postfach 50,D-7000, Stuttgart 1, 1991.
2)

7.2 Other useful references

1) *IC Microcontrollers*, Peatman, John B., Prentice Hall, New Jersey, 1998.

2) KVASER Controller Area Network pages : http://www.kvaser.com/can/index.htm

7.2.3 Websites

1) AnyBus Official Site : http://www.hms.se/

2) CAN in Automation : http://www.can-cia.de/

3) Dearborn Group, Inc. : http://www.dgtech.com/products/dpa.phtml

4) DeviceNet-ODVA Official Website :

http://www.odva.org/10_2/00_fp_home.htm

5) International Organization for Standardization homepage : http://www.iso.ch/

6) OSEK/VDX : http://www.osek-vdx.org/

7) Triangle Data : http://www.triangledigital.com/products/productscanbus.htm

8) Wesley Tang's Can Links : http://www.warwick.ac.uk/~esrpy/links.htm

9) Zanthic Products : http://www.zanthic.com/can4usbm.htm

SECTION 8. APPENDICES

8.1 Supporting Information

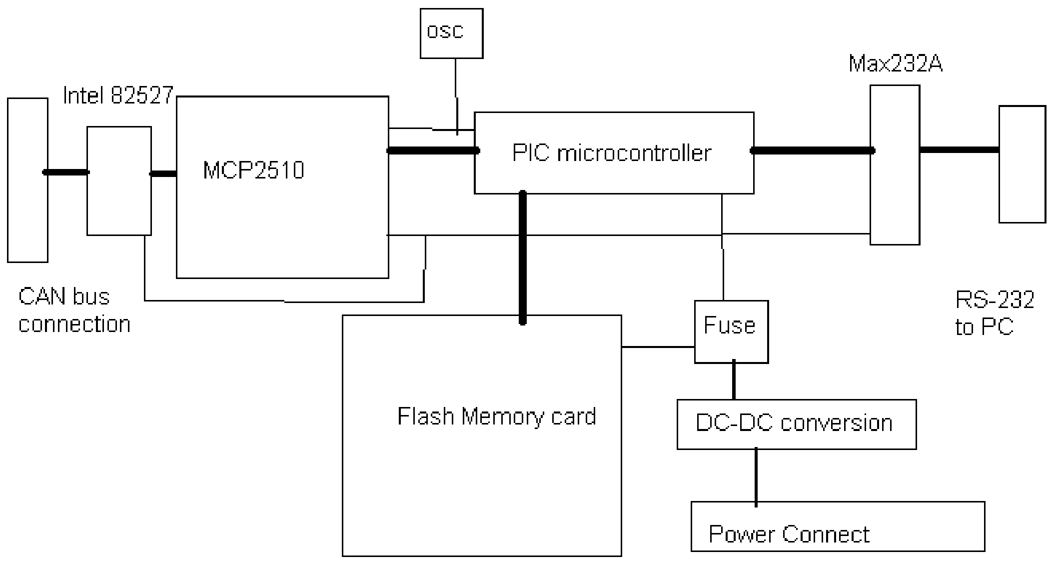


Figure 5: Sketch of CAN Bus Diagnostic Translator

8.2 Additional Block Diagrams

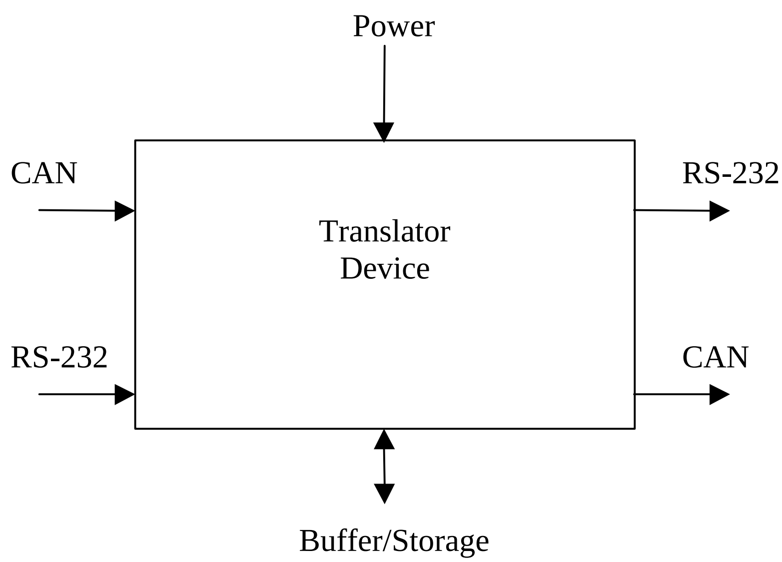


Figure 6. General Block Diagram

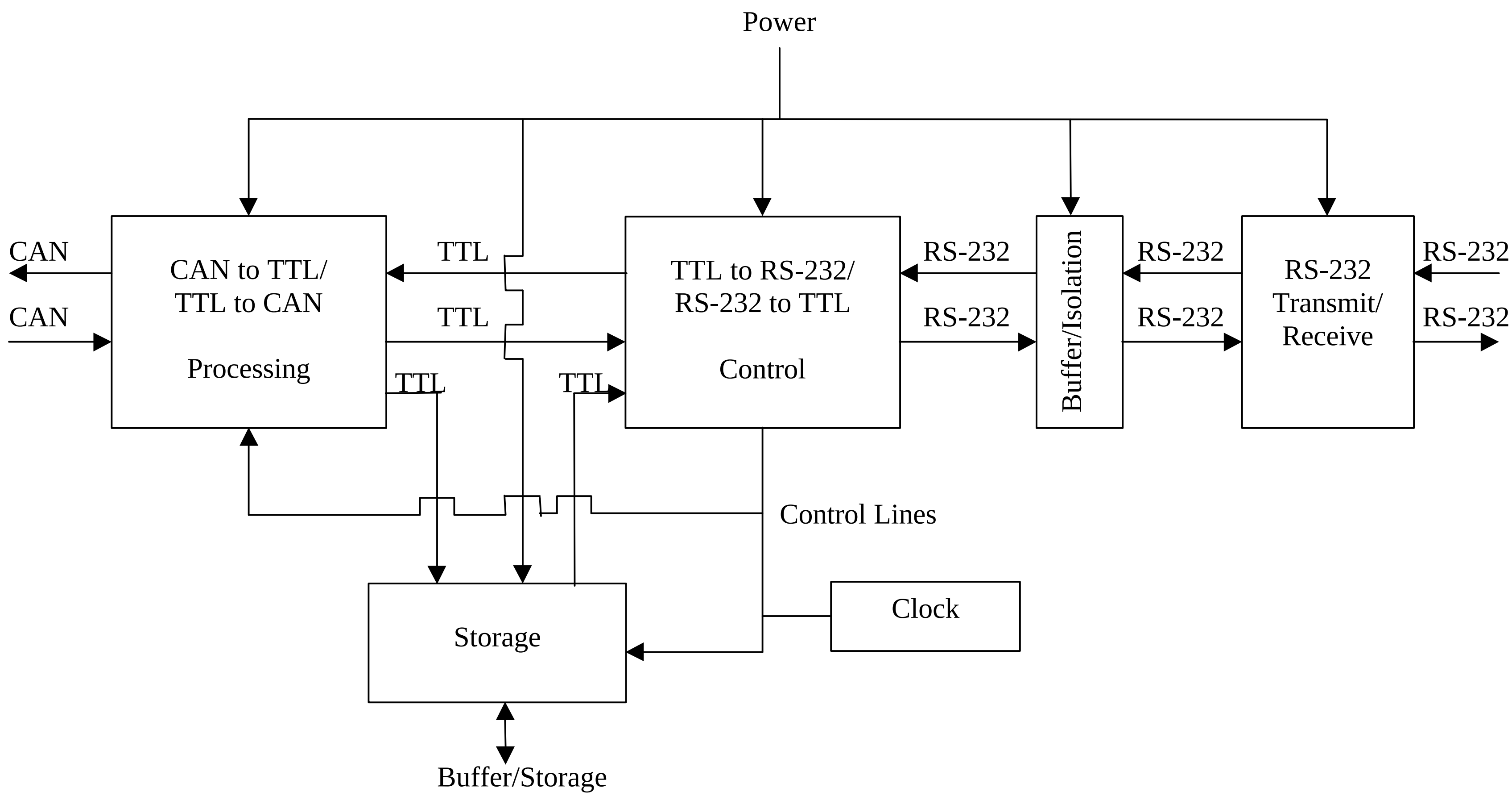


Figure 7. Expanded Block Diagram

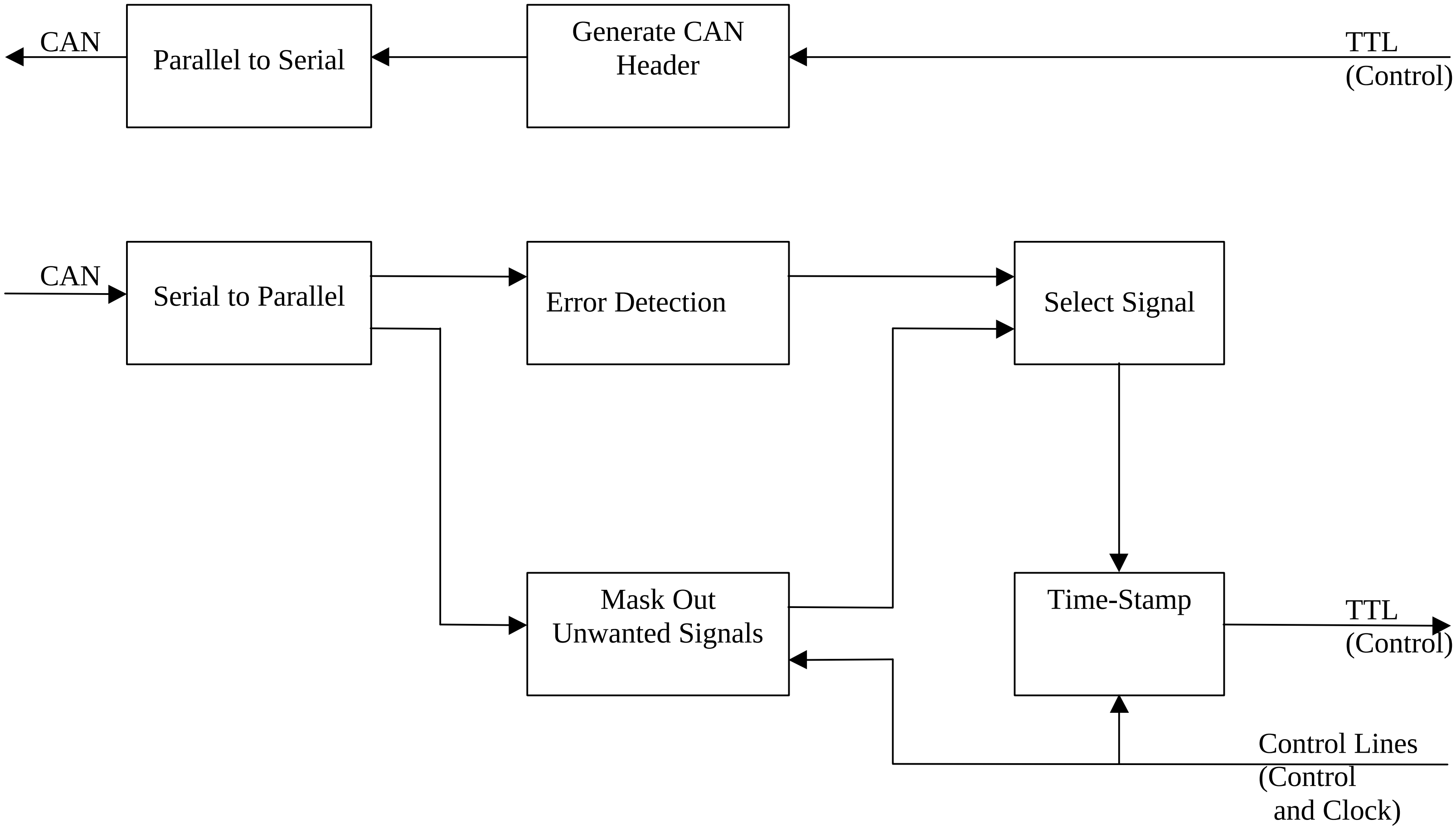


Figure 8. Detailed Block Diagram of Processing

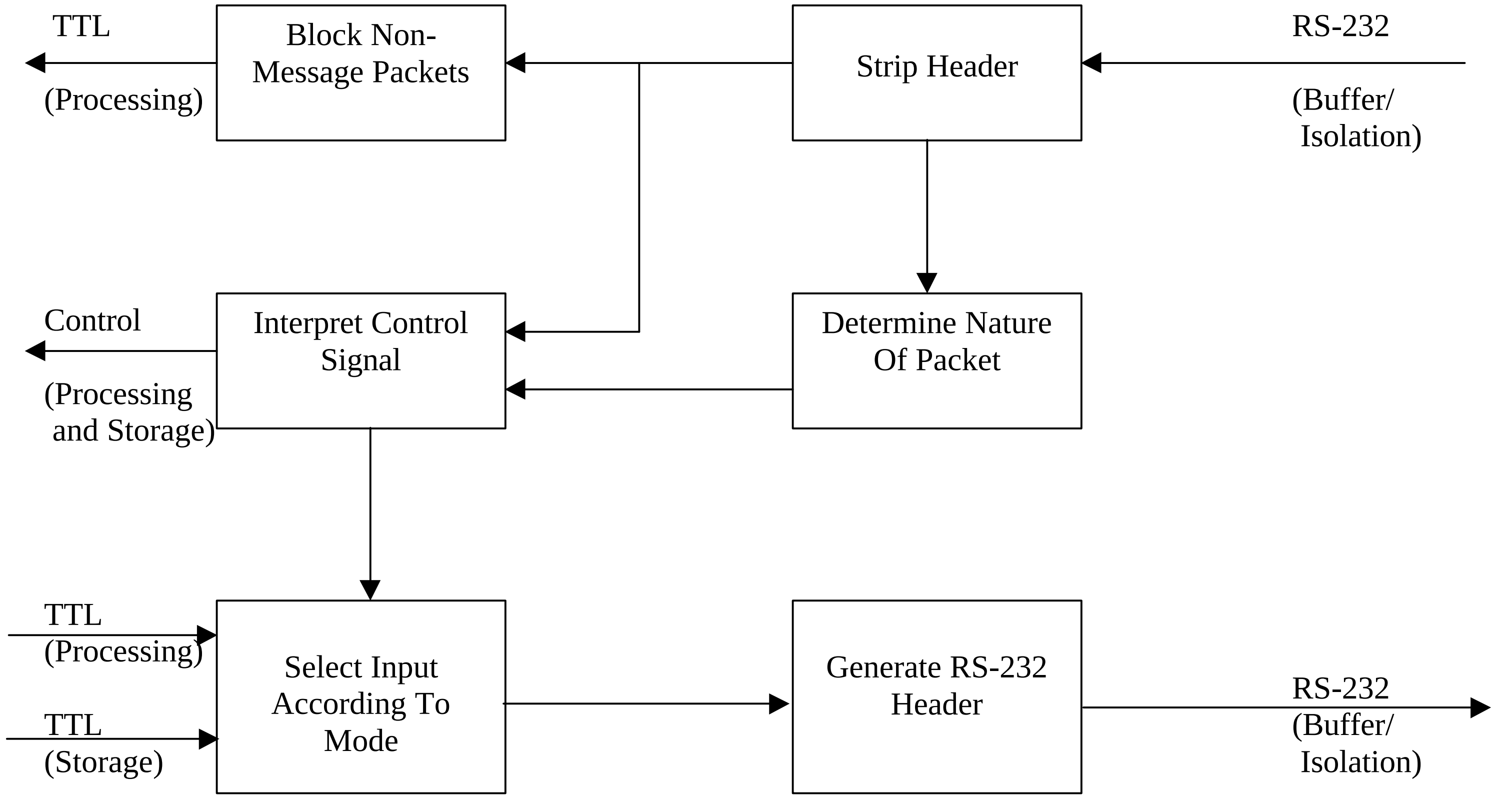


Figure 9. Detailed Block Diagram of Control

8.3 Determination of Flash Memory Requirements

CAN Bus Maximum Data Rate: 1Mbit/sec (~125KBytes/sec)

Assume 15 minutes of run time for a test run of the Formula Lightning Vehicle.

15 minutes * 60 sec/1 min = 900 sec.

900 sec * 125KBytes/sec = 112.5 MBytes

Assuming a 25% ‘duty cycle’ on the CAN BUS, where the ‘duty cycle’ is a combination of the bus not being utilized at 100% capacity and messages being filtered before storage:

112.5 MBytes * 0.25 = 28.125 MBytes.

So, a 32 MByte memory card should be sufficient for most test runs.