Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Development of a Cross-Compatible Micro-Avionics System for Aerorobotics

Taner Mutlu and Sertac Karaman, Savas Comak, Ismail Bayezit ‡,
Gokhan Inalhan*, Levent Guvenc**

Abstract— **In this work, we present a micro-avionics system structured around the Controller Area Network (CAN) Bus data backbone. The system is designed to be cross-compatible across our experimental mini-helicopters and ground vehicles, and it is tailored to allow autonomous navigation and control for a variety of different research test cases. The expandable architecture deploys a hybrid selection of COTS Motorola (MPC555) and Arm processor boards (LPC2294), each with different operating systems and coding techniques (such as rapid algorithmic prototyping using automatic code generation via Matlab/Real Time Workshop Embedded Target). The micro-avionics system employs a complete sensor suite that provides real-time position, orientation and associated time-rate information. As a part of the on-going fleet autonomy experiments, we present the design of a novel wireless SmartCan node. This wireless node allows seamless CAN Bus access of low-level sensor and operational data of other vehicles within close proximity.**

KEYWORDS: micro-avionics, robotics, autonomous vehicles, embedded electronics, sensor fusion, intelligent systems.

I. Introduction

Radio controlled air and ground vehicles serve as ideal testbeds for universities and research institutes for experimental demonstrations of unique control[3], [2], [6], navigation, communication and autonomy algorithms[7]. In this work, we present the design of a reconfigurable micro-avionics system that allows rapid-prototyping and testing of such aerorobotics experiments across a wide range of vehicle types. Our design is based on a research driven fundamental requirement to have a cross-compatible (i.e. the avionics can be used with minor modifications on different types of ground and air vehicles as depicted in Fig. 1) architecture that can support coordinated autonomy experiments across a heterogenous fleet of vehicles. Another major driver for this platform was that it should allow a collaborative hardware and software development environment, in which researchers can work on different topics (such as flight controls, image-driven navigation or multiple vehicle coordination algorithms) and could later integrate their work for flight and ground experiments with minimal hardware and software redesign concerns.

Fig. 1. The hardware-in-the-loop tests[7] of the micro-avionics core design. The avionics pod is placed on the 1/6 RC Humvee, which is used at the field tests. The servo lines are connected to the Vario Microhelicopter with the helicopter mathematical model running on a xPC target.

In comparison with elegant but monolithic micro-avionics architecture structured around different form factors such as single-board computers[3] or PC-104 stacks[5], we consider a scalable multi-processor architecture based on data bus backbones[2], [1]; CAN Bus and Ethernet. The current micro-avionics system embeds an expandable number of COTS (and cheap) processors. We use a PhyCore MPC555 board and an ARM LPC2294 board connected over the CAN Bus data-backbone (with ethernet available for data bus expansion). The micro-avionics system employs Garmin GPS receiver, 3 axis accelerometer/gyro Crista IMU, Honeywell Altimeter and Digital Compass as the primary sensors. In addition, it houses a wireless transceiver for communication with the ground control station. For a complete overview of the ground and associated supporting hardware, we refer the reader to [7].

‡Research Assistants, Controls and Avionics Laboratory, Istanbul Technical University

*G. Inalhan is an Assistant Professor at the Faculty of Aeronautics and Astronautics, Istanbul Technical University inalhan@itu.edu.tr

**L. Guvenc is a Professor at the Faculty of Mechanical Engineering, Istanbul Technical University guvencl@itu.edu.tr

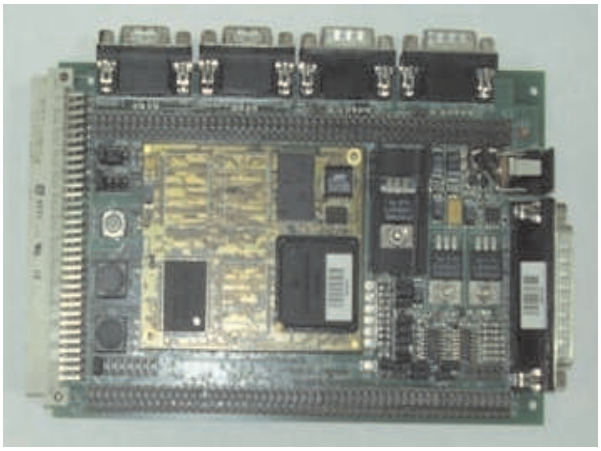

Fig. 2. MPC555 development board and the processor.

MPC555 is used with Matlab/Real Time Workshop to rapidly design and prototype algorithms using Simulink and the MPC555 embedded target automatic code generation feature. This fast-prototyping approach is a major distinguishing factor of our architecture in comparison with general CAN Bus based architectures[1]. Sensor integration to the Simulink environment is done through an in-house developed data parsing block-library for each device type. Field data collection and functionality tests of this core design using a 1/6 scale Humvee, and the hardware-in-the-loop tests of embedded control and way-point navigation algorithms on a Vario micro-helicopter has been succesful[7].

The ARM processor board, for which various custom driver codes (such as CAN and Serial Communication) was written run RTARM RTOS and has proved real-time operation success with computationally intensive tasks such as digital filtering. We have designed a generic SmartCan node using PIC 18F458 which allows us to interface serial data inputs/outputs to the CAN Bus. Through this, a novel wireless can node device is integrated. The device allows direct access(i.e. without communication overheads) of low-level sensor data of vehicles within close-proximity for to be used in coordinated fleet autonomy experiments.

In the next two sections, we review the basic hardware and software architecture of both the MPC555 core design and the final distributed micro-avionics system in corporating the LPC2294. In addition, we outline the design and the functionality of key subsystems such as pilot/operator autonomous control switch circuit, TPU and power interface board, CAN physical interfaces, SmartCan Nodes. We note that all of these hardware component are designed and manufactured in-house.

II. Core Design

Our core design is structured around the Phytec MPC555 board[1] shown in Fig. 2. MPC555 serves as the center piece CPU which communicates with the sensory devices (such as GPS, IMU, altimeter, digital compass), transmits data back and from the ground station through the wireless transceiver, and handle actuation operations with servo actuators.

One of the unique enabling features of MPC555 is that the Matlab/Real-Time Embedded Target allows automatic code generation of algorithms for rapid prototyping. In addition, the availability of drivers for serial and CAN Bus communications, MIOS functionality provides relief from coding of the physical drivers. However, its limited TPU usage capability and the necessary data-parse coding of each different type of sensor required a considerable customization for our purposes. Towards this end, we have developed a generic software library for the MPC555 processor in the Matlab/Real-Time Embedded Target environment. This generic library provides functional block diagrams that parse the sensor data and control the actuators using appropriate driving signals. In addition, initial sensor data fusion and the main control algorithms together with the path planner were also implemented on the same hardware using customized functional and algorithmic blocks.

Our physical design as shown in Fig. 3 is based on layers on the core processor. Three different layers are presented in this design. The Network Layer consists of the Wireless Transceiver unit, CAN based data logging unit and a radio receiver for pilot/operator control of the vehicle. Another layer is the physical interface layer which is used for interfacing some of the sensors to the core processor, and for switching the servo operation between the radio receiver (directed by the human operator) and the core processor in a safely manner. This is shown in Fig. 5. A third layer is built on the physical layer which contains the sensory devices and the actuator. This layer has the IMU, GPS, altimeter and the digital compass as well the servo actuators.

In the next subsections, we review the main characteristics of the MPC555 processor and the in-house developed custom libraries, and the different layers of the core design going over the hardware components : sensors, networking devices and the actuators.

A. MPC555 architecture and the in-house developed Real-Time Target Library

MPC555 is a very light weight and small size processing unit with several different on-chip features. MPC555

[1]This system has processing power of 40 MHz clock with performance measured as 52.7 MIPS at 40 MHz

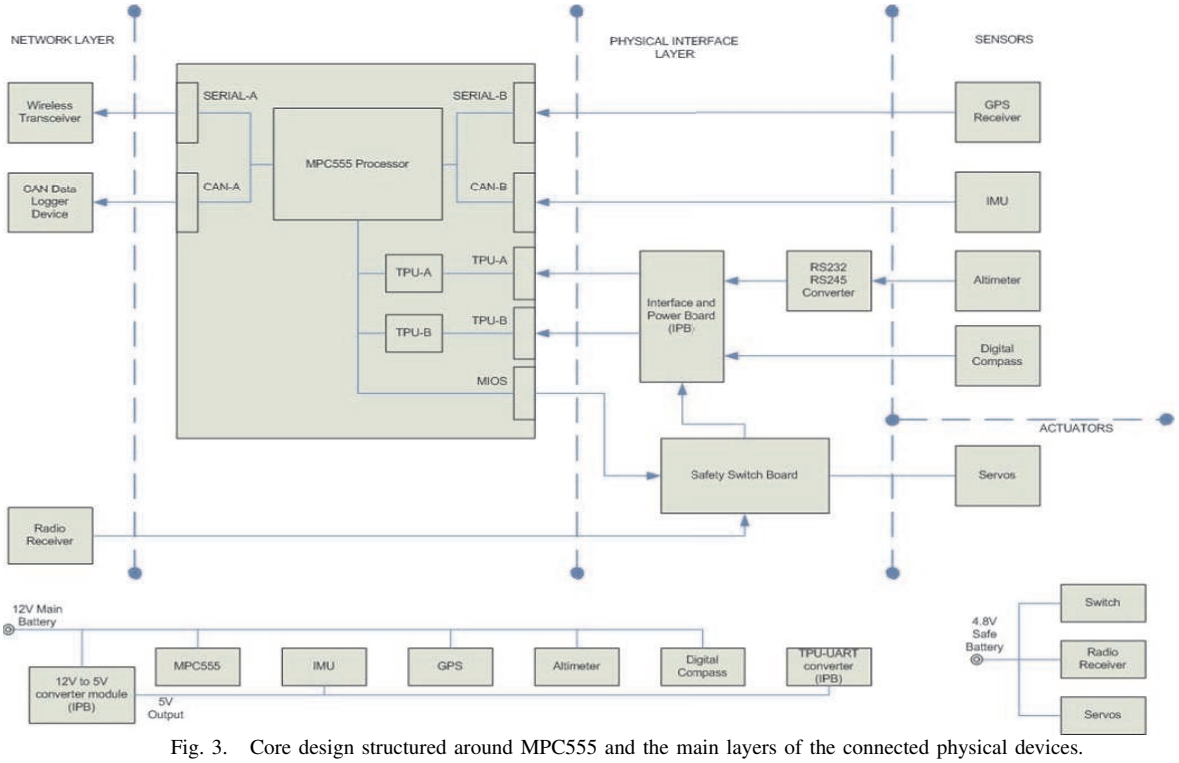

Fig. 3. Core design structured around MPC555 and the main layers of the connected physical devices.

processor was used with the development board provided by phyCORE. This development board houses the appropriate power connectors and converters, status LEDs, IRQ switch, reset switch and wake-up switch as well as connectors for RS232 and CAN connections. Our development board as shown in Fig. 2 also provides a programming interface which allows programming the processor using the PC parallel port.

The MPC555 processor has two on-chip RS232 drivers for which the development board provides the appropriate physical interface for RS232 communication. This feature is called the serial peripheral interface (SPI). One of the SPIs is a queued module which can handle up to 32 pre-programmed transfers and has 160-byte queue. Two on-chip CAN communication units which can be run at different data transmission rates are also provided by MPC555. Appropriate CAN controllers and physical layer including the connectors are provided with the development board. The CAN modules are fully compatible with the CAN 2.0A and CAN 2.0B standards.

Core processor system also includes two Time Processor Units (TPUs). Both of these units have 16 pins and each TPU pin can be configured for a function for which the code is provided in TPU RAM. TPU has a separate processor and clock, thus its functions do not overload the core CPU unit MPC555. One of the TPUs was mainly used for serial communication and the other for PWM acquisition in this work. The TPU based functions and the in-house developed driver code will be given in detail in the following sub-sections.

Another function of the MPC555 hardware is the Modular Input Output System (MIOS) unit. This unit provides eight dedicated pins for PWM generation functions and ten pins for some time acquisition functions. These time acquisition functions include acquisition of the pulse width or period of a digital signal. It also provides two 16-bit counter sub-modules which can be used for non interrupting counting action. MIOS also provides 2 parallel port I/O sub-modules.

MPC555 system provides two analog-to-digital converter modules with up to 16 analog input channels when using internal multiplexing. This number can be extended to 41 if external multiplexing is used. The A/D converter is a 16-bit converter and typical conversion time is 10 micro-seconds, i.e. 100,000 samples per second.

The MPC555 hardware is compatible with the Matlab/Real-Time Workshop. The code generation process for this design is thus handled in Simulink. The Simulink block diagram is then converted to C code using Embedded Target for Motorola MPC555 option in Matlab and the Real-Time Workshop of Matlab. Most of the aforementioned features of the MPC555 based core

design system are readily available as Simulink blocks in the 'Embedded Target for Motorola MPC555 Block Library'.

Using the basic hardware interface drivers, we have developed a block library for Simulink which is able to communicate with the sensors and parse the raw data appropriately to output meaningful sensor information with time stamps on the data if necessary. The block library developed as a part of this work also handles communication with the network layer devices as well as the servo actuators. The block library, as shown in Fig. 4,

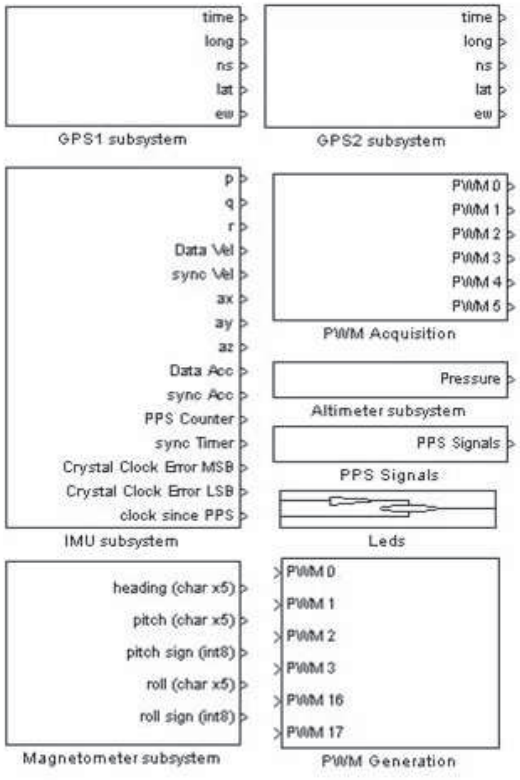

Fig. 4. The software library and drivers are developed in the native Simulink environment for plug-and-play sensor integration of rapid control algorithm prototyping.

provides two blocks for data acquisition from GPS unit and its parse operation. One of these blocks can acquire data from a GPS unit connected to the SPI port of the MPC555 system and the other can do the same job for a GPS unit that is connected to the TPU based serial ports. The in-house developed drivers for the TPU based serial ports output the time acquired from the GPS device, as well as the position and velocity of the receiver antenna. The block is configurable for different data sentences provided by the GPS unit. The library also contains one block for data acquisition and appropriate parsing for the IMU. The outputs of this block are the accelerations and angular velocities together with time stamps. The time stamp is defined according to the GPS Pulse-per-second (PPS) signal which is provided by the GPS when the GPS data is acquired in every one second. The Magnetometer subsystem block receives the data from the digital compass unit and parses it. The outputs are heading, pitch and roll angles. The final block associated with sensor devices is the altimeter block. This block provides the air pressure in psi units.

The PPS signals can also be connected to MPC555 - MIOS digital input channels for acquisition. This informs the CPU software upon the acquisition of the GPS signal. A block is provided for acquisition of this signal. The PWM acquisition block is used for acquisition of the PWM pulse width with TPU-B unit. This unit is connected to the radio receiver in the hardware design. Thus using this block the radio receiver signals can be read either for system identification purposes or for acquisition of human operator commands. There is also a PWM generation block which has inputs for desired pulse width. According to this pulse width the block generates the appropriate PWM signals with appropriate signal period to command the servo actuators. The final block is also present for flashing the development board LEDs continuously to be used for debugging purposes.

Using the block library provided in this work, the MPC555 based core design hardware can be used easily. The sensor information fusion code or the control loop can be implemented between the sensor parsing blocks and the actuation blocks in Simulink. Noting that Simulink is a good environment for controller development and development of algorithms like sensor fusion or path planning, this software structure is very suitable for rapid prototyping and controller, planner, fusion algorithm development when the provided block library is available. The real-time executable code provided by the Real-Time Workshop works in a common daisy-chain manner.

B. Network Layer

The network layer consists of a wireless transceiver, a flight data recording unit and a radio receiver [2].

The wavelink wireless transceiver from UKS is used for transmission of serial data to a ground station or another unmanned aerial vehicle. The connection with the device is through one of the SPI based RS232 ports of the device. The device allows point-to-point and point-to-multi-point data communication. The operation

[2]Wireless transceiver transmits data up to 19200 baud where as the CAN connection allows on-board data rates ip to 1M baud

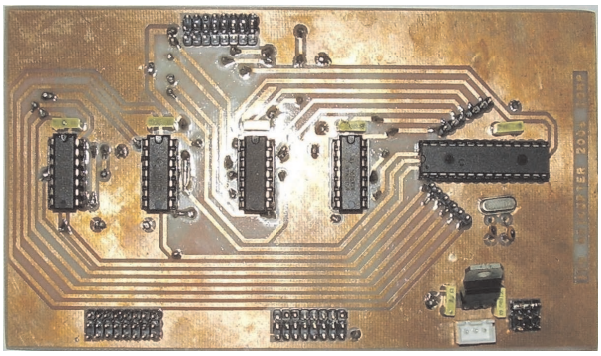

Fig. 5. The in-house designed and manufactured autopilot/operator switch board.

of the device is just like a null modem operation in between two communication points.

The flight data recording unit is used for acquiring data from the CAN bus and record it to SD and MMC compatible cards. The memory can be up to 2 GBs which allow recording for hours for this work. The recorded data can easily be taken to a PC for analysis. This CAN based recording unit is also used as a CAN-USB converter, which can then be used for better debugging or system monitoring purposes. We use the Kvaser CAN data logger as the flight data recording unit.

The radio receiver component is normally used for commanding the servo actuators under non-autonomous operation. In the autonomous system the RC receiver - Servo architecture is protected under a switching circuit which is able to switch between the non-autonomous and the autonomous operations.

The radio receiver output is basically the PWM signals used for driving the servo units. This signal is in 3.3V based digital logic. In order to acquire the human operator commands from the RC receiver, the 3.3V logic is converted to 5V logic within the TPU interface board and then fed to the TPU. Since one of the TPU functions is the acquisition PWM signal width, the acquisition process can be done easily without actual processing power loss in the main processor.

Receiver signals can be acquired in a system identification phase or a human operator command can be acquired for human collaborated control or decision making. In the system identification case, the signals acquired are time stamped and transmitted via the CAN Bus for being recorded by the flight data recorder unit. For human-machine collaboration, the acquired signals can be used within control and decision making algorithms. Still in both cases for data acquisition from the RC receivers the hardware and core software architecture is as described.

For better engineering practice, the receiver unit is connected to a safe battery together with the switching circuit board and the servo units. This safe battery is set apart from the main battery which is for the other sensory devices and the processor board itself.

C. Physical Interface Layer

This layer contains the TPU interface board the RS232-RS485 converter and the switch board. The TPU

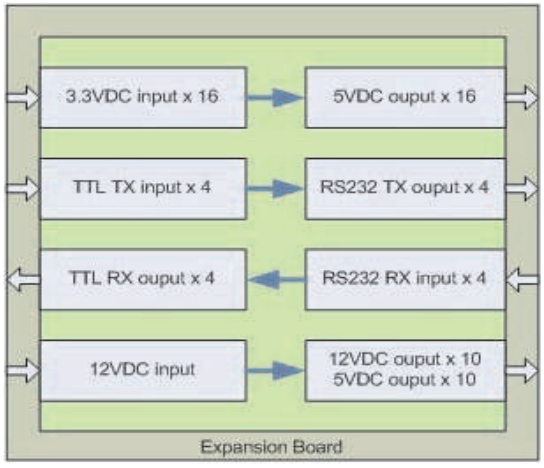

Fig. 6. The functional diagram of the TPU and power interface board.

interface board (the functional design is shown in Fig. 6) is used for conversion between TTL level serial communications signals and UART signals. Since the TPU units provide the serial communication signals in 5V TTL level, a conversion of these signals to 12V UART level is required. In the very same manner RS232 UART signals have to be converted to TTL level for acquisition of these signals by TPU units. For the conversion process, a single chip MAX232 IC was used in the TPU interface board. The TPU interface also includes converters from 3.3V to 5V. This process is required for conversion of 3.3V PWM signals acquired from the RC receiver to 5V logic from them to be acquired by the TPU unit. The interface board illustrated in the Fig. 4 uses sixteen 3.3VDC to 5VDC conversion channels, which is accomplished by two three-state octal buffer ICs(74HCT541). The four channel TTL to RS232 and vice versa conversion is accomplished by using two 5VDC powered multichannel RS232 drivers/receivers(MAX232). The 12VDC power comes from batteries and needs to be regulated to 5VDC in order to be used by most ICs and sensors. This board provides ten outputs for 12VDC and 5VDC each. In addition, a RS232-485 converter is used

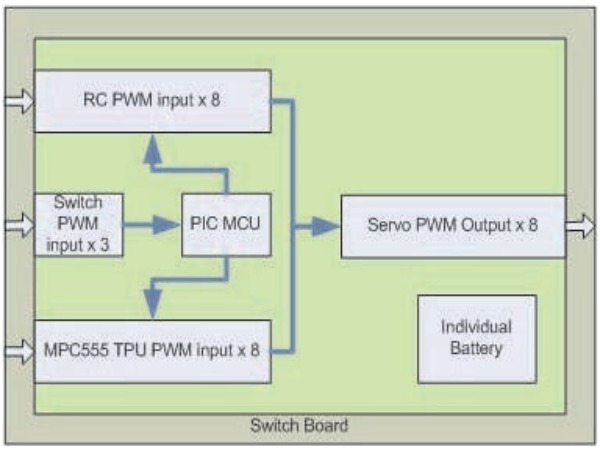

Fig. 7. The functional diagram of the autopilot/operator switch board.

for converting RS485 based serial signals provided by the altimeter unit to the RS232 type used by the TPU serial interface.

Eventhough the micro-avionic system is designed for autonomous flight tasks, it is very probable that unpredictable situations take form during flight tests, therefore a mechanism that will let the human factor intervene the platform during such conditions is a very critical functionality. For this purpose, a switch board (functional diagram and the board are shown in Figs. 5 and 7) is designed. An extended feature of the switch board is the capability of switching just any of the desired control channels. For example, if only the pitch control is desired to be switched to the human command, the switch board keeps all other controls at the embedded avionics system and switches the pitch control to the human command.

The Switch Board takes 11 PWM inputs from the RC radio transceiver and 8 inputs from the MPC555(or any other board) and directs the control inputs to the servos. 3 of the 11 PWM inputs is for the channel selection purpose and the remaining 8 inputs is for human control. The switching operation is established by a PIC 18F873 and the 74HCT125 three-state octal buffer is used as the interface of channel switches. In addition, for system identification and health-monitoring purposes, the signals going to the servo actuators are provided as another output. These signals can be used by MPC555 to estimate the servo position at any desired time.

D. Sensor and Actuator Layer

The sensor devices of the physical layer includes a three axis accelerometer/gyro IMU, a GPS unit, an altimeter and a digital compass.

As the GPS receiver, we use the readily available Garmin 15H. This device can track up to 12 satellites to determine the position and the velocity of the vehicle, and also outputs a PPS timing signal. This PPS signal is connected to the IMU for time stamping of IMU data. The NMEA sentences that will be transmitted by the GPS unit are user selectable through its custom interface. The GPS data is received approximately every one second.

For inertial acceleration and angular velocity measurements, we use Crista Inertial Measurement Unit (IMU)[3] provided by Cloud Cap Technology. This unit is able to output the acquired data both via a RS232 based serial interface and a CAN interface. It also provides internal oversampling and averaging of the oversampled data automatically before transmission. The data transmission and oversampling rates are user selectable. The IMU is also able to time stamp the acquired signals via the PPS signal acquired from the GPS unit. A data update rate of 50 Hz is used in this work with an oversampling rate of 20.

The altimeter unit used, Honeywell HBP, is able to provide both temperature and pressure measurements. Thus the pressure measurements can be compensated according to the temperature. We use the factory setting data update rate of 5 Hz. The data update rate for this device is user selectable.

The digital compass used for inertial orientation aiding is the Honeywell HMR 3300. This unit provides three axis measurement of orientation relative to the earth magnetic field. The orientation information of the heading angle is of 360 degrees and fully covers the range whereas the pitch and the roll are of 60 degrees. The data update for this device is 8 Hz.

E. Synopsis of the Core Design

The outdoor functional tests of the core design on a 1/6 scale RC Hummer have been succesful, achieving all the sensor data collection goals[7]. In addition, we were able to embed high-level control and way-point navigation algorithms at the hardware-in-the-loop tests using the Vario micro-helicopter. During these tests,we have observed (as expected) that the rapid code development process can result in non-customized real-time code which then results in inefficient usage of computational power. In this form, the capability of such cheap COTS embedded targets can be utilized to either (a) full sensor

[3]IMU Gyro Range: 300 degree/sec, Gyro Resolution: 0.009 degree/sec, Accelerometer Range: $\pm$ 10g, Accelerometer Resolution: 0.3mg

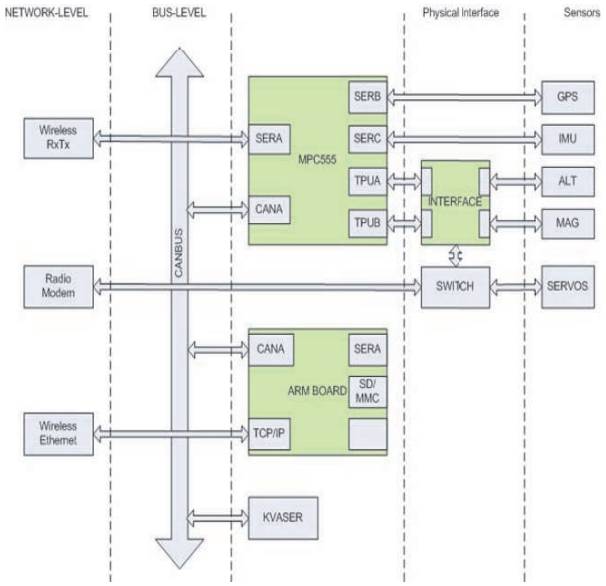

Fig. 8. Distributed multi-processor design including the ARM processor board which connected to the core design via CAN Bus.

data fusion with only fundamental control law algorithmic implementations or (b) complete high-level control and coordination algorithm implementations with only bus-based (such as CAN Bus) standardized sensor data flow and fusion.

Our distributed multi-processor design allows this core design to be utilized in its unique and full potential as a rapid-prototyping and integration board. This new design off-loads other computationally intensive tasks to processors with direct interrupt controls and multi-threading. In addition, the integration of SmartCan nodes, which convert serial data into CAN Bus format, allows us to integrate sensors directly via CAN Bus and test unique concepts. One such concept tested is the usage of wireless SmartCan nodes that allow other vehicles to plug into the low-level sensor data of the platform without hefty communication overheads. This device provides an enabling technology for not only coordinated behavior in close proximity operations, but also the possibility of directed controls of other devices.

III. Distributed Multi-Processor Design and SmartCan Expansion

The distributed multi-processor design as shown in Fig. 8 is structured around CAN Bus line (with additional flexibility to include an Ethernet bus line also) providing the ability to interoperate independent of processors with different functionalities, native operating systems and thus coding standards. This also streamlines the micro-avionics allowing the reconfigurability for different sensor sets through sensor customization via SmartCan modules.

A. SmartCan Nodes

The SmartCan node design uses PIC 18F458 to direct serial data flow to and from the CAN Bus line. Once the TTL signals are converted by MAX232 and processed by the PIC, PCA82C250 is used for transition from the CAN protocol controller to the physical bus CAN High and CAN Low outputs. In addition, the node is equipped with 12V-5V converter and also allows A/D processing for analog and digital signals. The operational functioning of the SmartCan node is shown in Fig. 11. In our tests, we have created the wireless data transmission SmartCan node using a UTR-C10M transceiver.

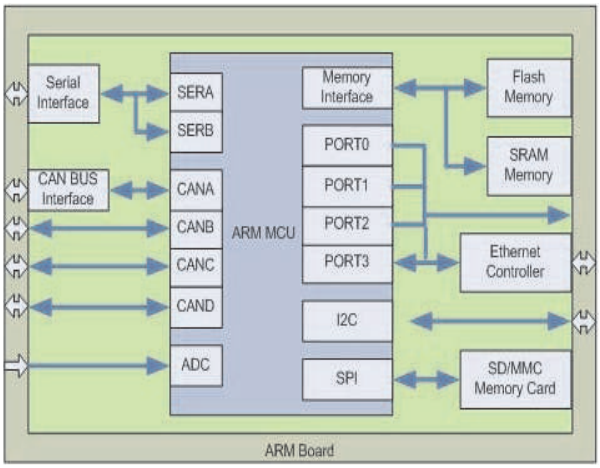

Fig. 9. Arm board functional diagram.

B. Arm Processor

The ARM Board that we use is a ARM 7 family Philips LPC2294 microcontroller based prototype development board as shown in Fig. 9. The MCU can run up to 60 Mhz speed and process up to 60 MIPs utilizing the pipeline feature. In addition, the board supports our usage of Real Time Operating Systems(RTOS) such as VxWorks, QNX, uClinux and freeRTOS. This ability has provided us with capability to do extensive multitasking operations and create a custom structured organization of the software.

The Ethernet controller chip on the bottom side of the board allows TCP/IP communication usage, as well as the CANBUS protocol usage made possible with the four internal CAN controllers(in the ARM7 MCU) and a CAN physical interface IC on the board. For the other remaining three CAN channels, we have designed and manufactured an in-house Can Physical Interface board shown in Fig. 10. The Arm development board

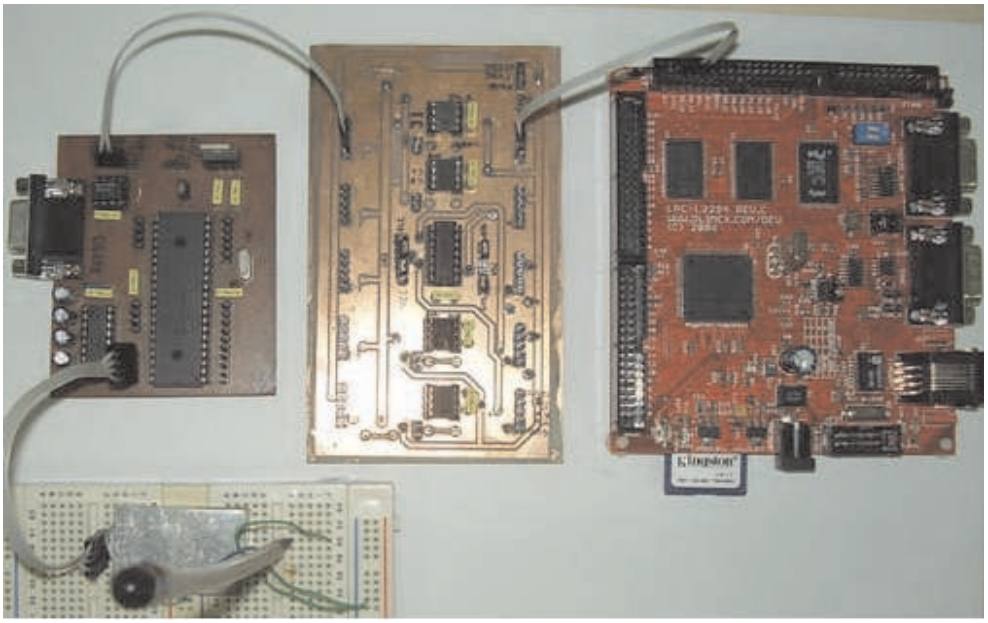

Fig. 10. Arm board featured with in-house designed and manufactured Can Physical Interface board and the SmartCan Node connected to UDEA wireless transceiver before a system functionality check.

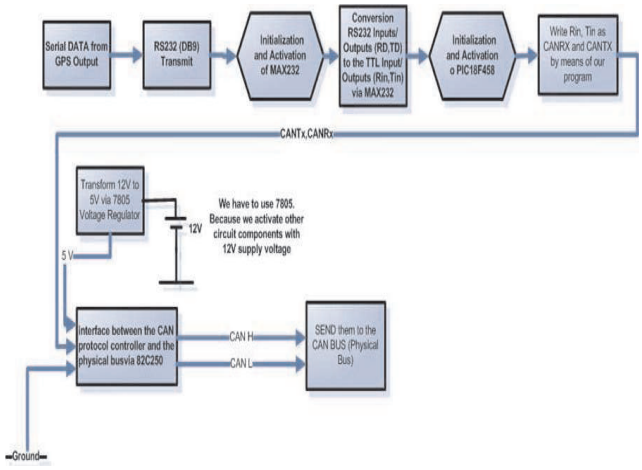

Fig. 11. SmartCan Operation Module for GPS receiver

shown has been used and tested for prototyping many of our software routines including digital Kalman Filtering, GPS receiver NMEA protocol parsing, RF transceiver receive/transmit, and motor and servo controls.

IV. Conclusions and Future Directions

Our current research focuses on extending the use of the SmartCan nodes in the distributed architecture design and on experimental field and flight tests of the micro-avionics system.

V. Acknowledgments

The authors gratefully acknowledge the contribution of Dr. Erdinc Altug for choosing MPC555 as the core-processor and Murat Demirci for developing the first tests of MPC555 sensor integration with Crista IMU. This work is partially supported by DPT HAGU program administered by the ITU ROTAM.

References

[1] J. Elston, B. Argrow, and E. Frew,"A Distributed Avionics Package for Small UAVs," *in AIAA Infotech at Aerospace*, Arlington, VA, September 2005.

[2] E.N. Johnson, and D.P. Schrage,"The Georgia Tech Unmanned Aerial Research Vehicle: GTMax," *in Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[3] V. Gavrilets, A. Shterenberg, M. A. Dahleh and E. Feron, "Avionics system for a small unmanned helicopter performing aggressive maneuvers,"*in IEEE Aerospace Electronics Systems magazine*, September 2001, Vol 16, 9.

[4] J. Evans, G. Inalhan, J.S. Jang, R. Teo, and C. Tomlin, "Dragonfly: A Versatile UAV Platform for the Advancement of Aircraft Navigation and Control," *in the Proceedings of the 20th IEEE Digital Avionics Systems Conference*, October 2001.

[5] J. S. Jang and C. Tomlin,"Design and Implementation of a Low Cost, Hierarchical and Modular Avionics Architecture for the DragonFly UAVs," *in Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, AIAA Paper Number 2002-4465, August 2002.

[6] D. H. Shim, H. J. Kim and S. Sastry, "Hierarchical control system synthesis for rotorcraft-based unmanned aerial vehicles," *in the Proceedings of AIAA Guidance, Navigation, and Control Conference* , 2000

[7] S. Karaman, M. Aksugur, T. Baltaci, M. Bronz, C. Kurtulus, G. Inalhan, E. Altug and L. Guvenc, "Aricopter : Aerobotic Platform for Advances in Flight, Vision Controls and Distributed Autonomy,"*in Proceedings of IEEE Intelligent Vehicles Symposium, 2007*.