International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:9, No:12, 2015

Development of Monitoring Blood Bank Center Based PIC Microcontroller Using CAN Communication

Kaiwan S. Ismael, Ergun Ercelebi, Majeed Nader

***Abstract*—**This paper describes the design and implementation of a hardware setup for online monitoring of 24 refrigerators inside blood bank center using the microcontroller and CAN bus for communications between each node. Due to the security of locations in the blood bank hall and difficulty of monitoring of each refrigerator separately, this work proposes a solution to monitor all the blood bank refrigerators in one location. CAN-bus system is used because it has many applications and advantages, especially for this system due to easy in use, low cost, providing a reduction in wiring, fast to repair and easily expanding the project without a problem.

***Keywords*—**Control Area Network (CAN), monitoring blood bank center, PIC microcontroller, MPLAB IDE.

I. Introduction

THERE is three main components of blood RBC, platelet, and plasma each component should be stored in different refrigerators because of the temperature limit's range. For example, RBC needs to be stored at a temperature 20 to 24°C, Plasma needs a temperature from -18 to -20°C, and Platelet needs an environment from 1 to 6°C. Due to security location in the blood bank hall, it is difficult to monitor the temperature of each of the refrigerators separately. This hardware development solves the problem by monitoring all refrigerators in one place and sited each refrigerator in a limited temperature range depends on a blood component permitted temperature. The developed system includes four nodes, which are named as RBC, plasma, platelet, and display modes. The nodes of RBC, plasma, and platelet have a thermometer to read the temperature of each refrigerator and send them to the LCD display. In addition, each node sends alarm address of each refrigerator to the LCD display when override temperature is occurring. To implement an efficient communication between the nodes, the CAN bus protocol is employed.

The CAN bus is developed by Robert Bosch in the 1980's and is used as a multi-master protocol that can have a maximum signal rate of 1Mbps [1]. The idea of CAN came about to solve an ever increasing demand for the multiple Electronic Control Module (ECM) networking. CAN uses a smaller packet sizes to deliver messages throughout the CAN bus. For this reason, messages like temperature could be sent from one node to another with accuracy and with little cost. CAN is an International Standardization Organization (ISO) defined serial communications bus originally developed for the automotive industry that replaces the complex wiring harness today. The specification calls for high immunity to electrical interference (EMI). For these features, CAN's popularity is used in a variety of industries including building automation, medical, and manufacturing [2]. The CAN communications protocol, ISO-11898 describes how information is passed between devices on the network. Actual communication between devices connected with the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the Data Link Layer (DLL) and the Physical Layer to ensure compatibility between CAN transceivers and only the transceivers [3].

We designed the employed principles to monitor the blood centers by using MPLAB and C code. PIC microcontrollers are used to implement the hardware architecture of each node. The nodes that have a thermometer to read the temperatures in the blood centers, communicate with the display node through in implementing CAN bus system.

Kaiwan Saber Ismael and Ergun Ercelebi are with the Electrical and Electronic Engineering, Gaziantep University, Turkey (e-mail: kaywaneng@gmail.com, ergun.ercelebi@gmail.com).

Majeed Nader is with the Electrical and Computer Engineering Department, Wayne State University, Michigan, USA (e-mail: majeed@wayne.edu).

II. Hardware Architecture

The hardware module is consisting of four main nodes, each node in different places and collects all refrigerator temperatures in one place. Also each node sends data to the bus.Our hardware is simulated in a one card and consist it Peripheral Interface Controller (PIC18F458), Control Area Network (CAN) Transceiver, Voltage Regulator (LM7805C), Temperature sensor LM35, Liquid Crystal Display (LCD) 2X16 (2 line and 16 character display) Buzzer, LED,7 channels Darlington Sink Driver (ULN2003APG), Push button switch.

A. Peripheral Interface Controller (PIC)

PIC microcontroller is a device that has internal memory; Read Only Memory (ROM), Random Access Memory (RAM), Central Processing Unit (CPU) and Input/output (IO) ports. All of these parts are built on a one chip called a microcontroller. PIC basically has a few KB of ROM, 256 or less bytes of RAM, 256 bytes of Electrically Erasable Programmable Read Only Memory (EEPROM) and several analog and digital IO lines. Microcontroller also has many types and different pins such as 28, 40 and 44 pins depend on the function of the microcontroller. For example, PIC18F458 has five ports, port A, B, C, D and E. It also has Typical 1536 Bytes Of Data Ram, 256 Bytes Of EEEPROM And Operating frequency 40 MHz, 34 Input /output Pins, 4 Timer, 8 Channel (10-bit) ADC (analogue to digital converter) [4].

B. Controller Area Network (CAN) Transceiver

The MCP2561 [5] shown in Fig. 1 is a high-speed CAN, fault-tolerant device that serves as the interface between a CAN protocol controller and the physical bus. The MCP2561 provides differential transmit and receive capability for the CAN protocol controller and is completely compatible with the ISO-11898 standard physical layer, including 5V requirements. It will operate at speeds of 1 Mb/s. Typically; each node in a CAN system must have a device to convert the digital data generated by a CAN controller to data suitable for transmission over the Bus cabling as shown in Fig. 2. It also provides a buffer between the CAN controller and the high voltage spikes that can be generated on the CAN Bus.

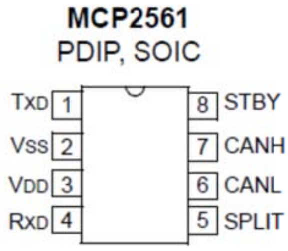

Fig. 1 CAN Transceiver

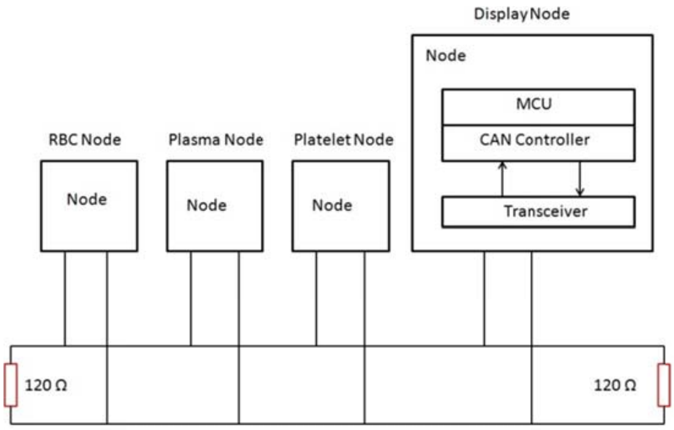

Fig. 2 CAN Bus Block Diagram

The main features are it supports 1 Mb/s operation. It is suitable for 12V and 24V systems. It is a low current standby operation. There is protection against damage due to short circuit conditions (positive or negative battery voltage). Also, there is protection against high-voltage transients and it has Automatic thermal shutdown protection. Up to 112 nodes can be connected.

C. LM7805C Voltage Regulator

The operating voltage range of PIC18F458 is 2.0V to 5.5V. So LM7805C which is a +5.0V 1A voltage regulator, is used. It is a linear voltage regulator that produces a relatively constant output voltage of +5VDC. There is an input pin, which must generally be greater than +7VDC, a ground pin, and an output pin as shown in Fig. 3.

D. Analog to Digital Regulator

In physical world parameters such as temperature, pressure, humidity, and velocity are analog signals. A physical quantity is converted into electrical signals. We need an analog to digital converter (ADC), which is an electronic circuit that converts continuous signals into discrete form so that the microcontroller can read the data [6]. As a PIC microcontroller also have built-in ADC, so the analog circuitry can be connected directly to the PIC Microcontroller unit. However, it depends on how much data unit supported in electrically erasable programmable read only memory (EEPROM) [7].

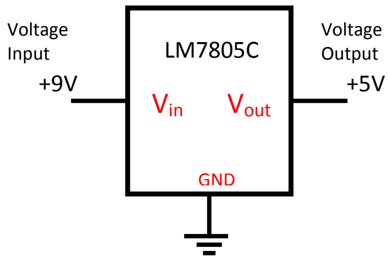

Fig. 3 Voltage Regulator Circuit

E. Temperature Sensor LM35

ThLM35 is an integrated circuit sensor [8]. Shown in Fig. 4 that can be used to measure temperature with an electrical output proportional to the temperature (in °C). It measures temperature more accurately than a using a thermistor. The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified. It has an output voltage that is proportional to the Celsius temperature. The scale factor is. 10V/°C. The LM35 does not require any external calibration Or trimming and maintains an accuracy of +/-0.4°C at room temperature and +/- 0.8°C over a range of 0°C to +100°C. Another important characteristic of the LM35 is that it draws only 60 micro amps from its supply and possesses a low self-heating capability. The sensor self-heating causes less than 0.1°C temperature rise in still air. The sensor has a sensitivity of 10mV /°C.

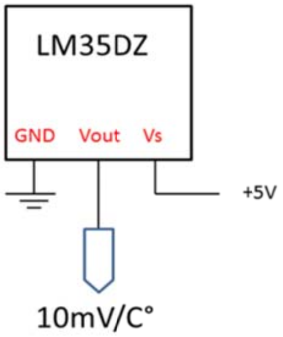

Fig. 4 LM35 Temperature Sensor

III. System Configuration

This system is made up of four CAN nodes. One node (called DISPLAY node) requests the temperature periodically and displays it on an LCD. This process is then repeated continuously. The second, third and fourth node reads the temperature of all refrigerators from an external semiconductor temperature sensor. This will be linear active sensor that converts temperature to an analog voltage. Also each node is far from together in a different place. The chosen microprocessor will be the cost Efficient is seeing that the controller consists of an already included CAN module. This controller will be the PIC18F series micro which contains the following CAN features:

- Complies with ISO CAN Conformance Test,
- Message bit rates up to 1Mbps,
- Conforms to CAN 2.0B Active Spec with:
- 29-bit Identifier Fields,
- 8-byte message length,
- 3 Transmit Message Buffers with prioritization,
- 2 Receive Message Buffers,
- 6 full, 29-bit Acceptance Filters,
- Prioritization of Acceptance Filters,
- Multiple Receive Buffers for High Priority,
- Messages to prevent loss due to overflow,
- Advanced Error Management Features.

The system is designed such that each node will send and receive messages across a network. Given we chose ECS Inc's 4 Mhz Oscillator (ECS) our nominal Baud Rate will be 25Kb/s. Nodes will be networked together such that each node will perform some action given the voltage output of a temperature sensor. The temperature sensor of choice is the LM35DZ which provides us a10mV per degree Centigrade output with no calibration necessary. By using this sensor to read the output signal into PORTA, pin RA1of the one of the PIC18F458 [9]. Controller's Analog to Digital Converter. A final PIC18F458 microcontroller will be used to display the temperature in °C. The CAN Module Bit Timing [10] is made up of non-overlapping segments. Each of these segments is made up of integer units called Time Quanta (TQ). The Nominal Bit Rate (NBR) is defined as the number of bits per second transmitted by an ideal transmitter with no resynchronization.

- Fosc = 4 MHz.
- Tosc = 1/4M = 0.25 Micro S
- Baud rate = 25 Kbit/s
- If BRP = 4
- TQ = 2 X (4+1) x Tosc = 2.5 Micro S
- TBit = 1/25k = 40 Micro S
- TBit = TQ X 40/2.5
- TBit = 16 TQ

Nodes must have same Nominal Bit Rate (number of bits per second).

- The Nominal Bit Time is defined as: TBIT = 1/Nominal Bit Rate
- Synchronization Jump Width (SJW): 1-4
- Baud Rate Prescaler (BRP): 1-64
- Propagation Time Segment (Prop_Seg) 1 - 8
- Phase Buffer Segment 1 (Phase_Seg1) 1- 8
- Phase Buffer Segment 2 (Phase_Seg2) 1-8

$$\text{Prop Seg} + \text{Phase Seg 1} \geq \text{Phase Seg 2}$$

$$\text{Phase Seg 2} \geq \text{Sync Jump Width}$$

These timing bits for our project were found to be:

- SJW = 1;
- BRP = 4;
- Phase_Seg1 = 4;
- Phase_Seg2 = 3;
- Prop_Seg = 3

$$\text{Prop Seg} + \text{Phase Seg 19} \geq \text{Phase Seg 2}$$

$$\text{Prop Seg} + \text{Phase Seg 1} = 7$$

$$\text{Phase Seg 2} \geq \text{Sync Jump Width}$$

IV. The Implemented Algorithms

This project is a four node CAN bus project, which are Display, Plasma, platelet, and RBC node as shown in Fig. 5.

A. Display Node

The display node is consisting of a PIC18f458, a 4 MHz Oscillator, a transceiver MCP2561, 7channel Darlington Sink Driver ULN2003APG is connected to the PORTE of PIC, Buzzer, LED and an LCD is connected to the PIC via port D. All the components interface with each other and work based on a specified algorithm. Using C-Language that is compiled on MPLAB v.92 we have programmed the PIC. The PIC18f458 has one build in CAN module and CAN controller. This facilitates the CAN project because there is an internal interface between the CAN module and the CAN controller. The shows the circuit diagram of the Node.

The first thing must be done is initializing the CAN module. Using the specified CAN initializing parameter and written code functions in C, the CAN module is initialized. Then it should be put the CAN module in the CONFIG mode and must be set some parameters in the masks and filters. We assign the filter B2-F3 for this node and set the band pass as 6 for RBC, 7 for platelet and 8 for plasma node. That means this CAN module passes any messages with identification of 6 during setting filter to 6 and so on. Then the Module is put back to NORMAL mode, which means that it is ready to work.

Initializing the LCD so as to be ready to get the data and show it. After initializing the LCD, the node should send a message to the bus that contains character "T", as a symbol of temperature, with a specified identifier, for example 65 in our case. Reading the messages on the bus, this module passes all messages that have 6 during filter setting as 6 identifiers. The messages with ID=6 contain the data of temperature that has been sent by another node, which is RBC (Red Blood Cell) node in our case and so on. This node sends the character "T" to the bus, reads the temperature data on the bus every second, and shows the Temperature in a suitable position on the LCD continuously which is the updated temperature from the Temperature Node.

All the necessary procedures to achieve the display node algorithm are shown in Fig. 7.

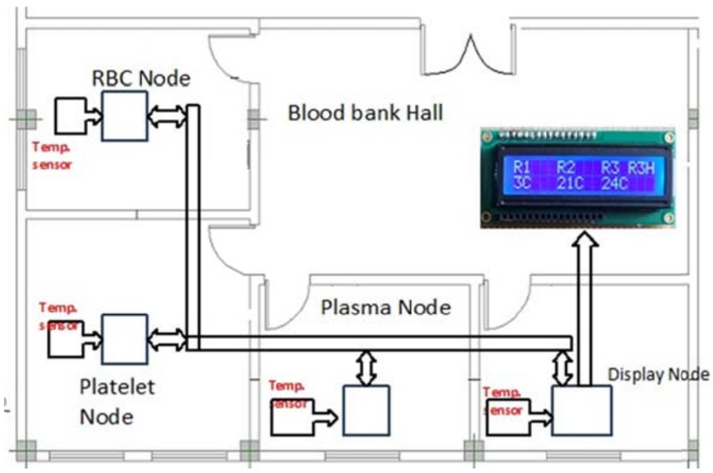

Fig. 5 Blood Bank Center designed layout

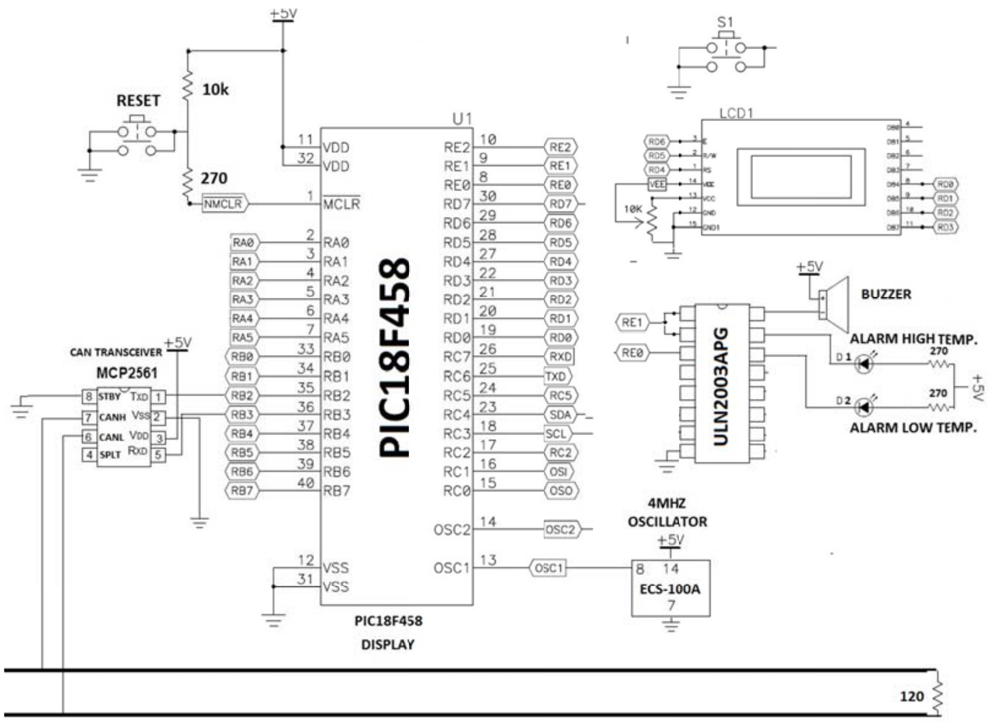

Fig. 6 Display Node Schematic Diagram

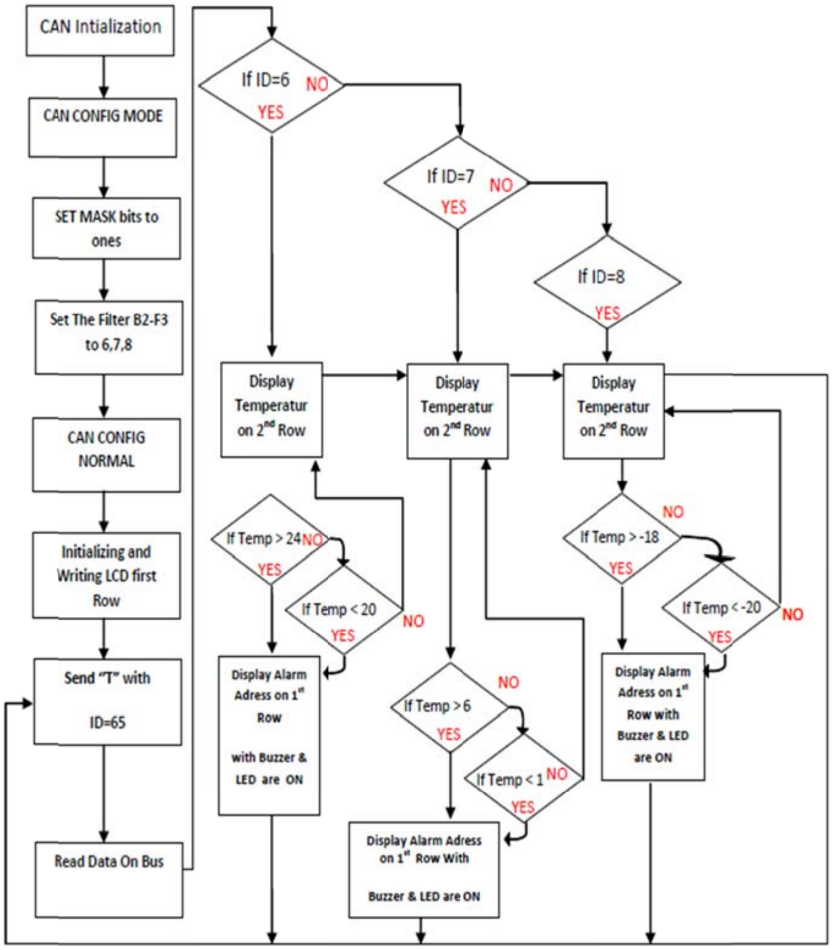

Fig. 7 Flow Chart For CAN Based Display Node

B. Red Blood Cell (RBC), Platelet and Plasma Node

This node has the following components: a PIC18f458, an LM35 temperature sensor, a 4 MHz Oscillator, and an MCP2561 transceiver. The temperature sensor is connected to the Analog to Digital Converter (ADC) which is a module inside the PIC through Port A (RA1). The structure of the node is illustrated in Fig. 8.

We configure the module in the similar way we do for the Display Node except the filter that need to be set to 65 so that it passes the messages on the bus that are being sent by the display node. The ADC module should be configured and turned on so as that it can convert the analog input signal that comes from the temperature sensor. The temperature sensor continuously reads the actual temperature and sends the temperature as a voltage signal to ADC input pin which is RA1 in my case. The node reads the messages on the bus and filters all the messages except the ones have 65 as an identifier. As soon as the node reads the message with the correct identifier, it checks if the data contains the character "T". If there is a character "T" in the message, the RBC node reads the output of the ACD and builds a CAN packet message that contains the actual temperature data and sends it to the bus with the identifier of 6. Similarly, the platelet and plasma nodes build the messages containing the temperature information and send them to the CAN bus with difference identifier. We used the identifiers 7, 8 for the platelet and plasma nodes respectively. The reading bus and sending temperature to the bus is a loop procedure that occurs every second. Fig. 9 shows the flowchart that explains the algorithm of the RBC, platelet, and plasma node.

V. Results and Discussion

1) Fig. 10 shows the result of the designed system after collecting and running the system. The LCD node that sends the data to the bus, collecting data on the bus and then shows the three nodes Temperature in Celsius in the second row of the LCD. While being shows the name of each Refrigerators and the alarm address on the first row of the LCD. The temperature shown on the LCD changes based on the data, which is generated and sent to the bus by the RBC, platelet and plasma node. The temperature sensor continuously reads the temperature and the temperature node generates the CAN packet data and put

it on the bus every second. Therefore, a temperature that is shown on the LCD gets updated every second.

2) The alarm (Address on the LCD, Buzzer and LED) works properly according to the designed algorithm. It turned ON when the temperature greater or less than the limited range and it turned OFF when the temperature remaining in the normal range.

3) We can see the data flow on the bus using an oscilloscope as it is illustrated in Fig. 10. We can see the data as a pulse train on both CAN high and CAN low. The upper part which is in yellow is CAN high, while the lower part which is in blue color is the data on CAN low. Notice that there is the same data on CAN high and low part, except they are negative to each other.

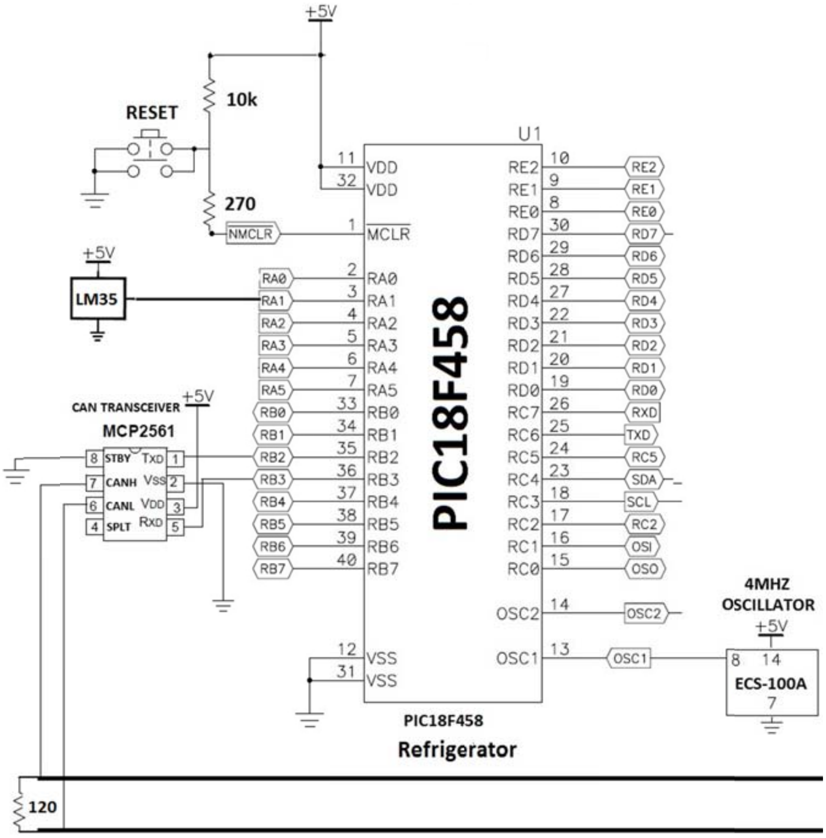

Fig. 8 RBC, Platelet and Plasma Node schematic diagram

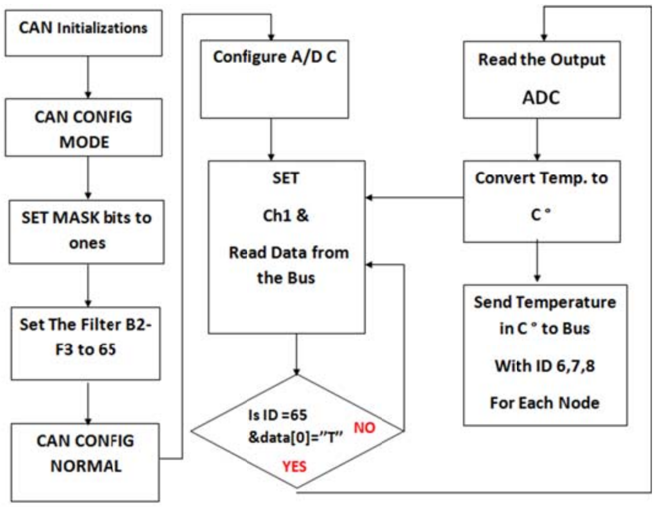

Fig. 9 Flow Chart for CAN Based RBC, Platelet and Plasma Node

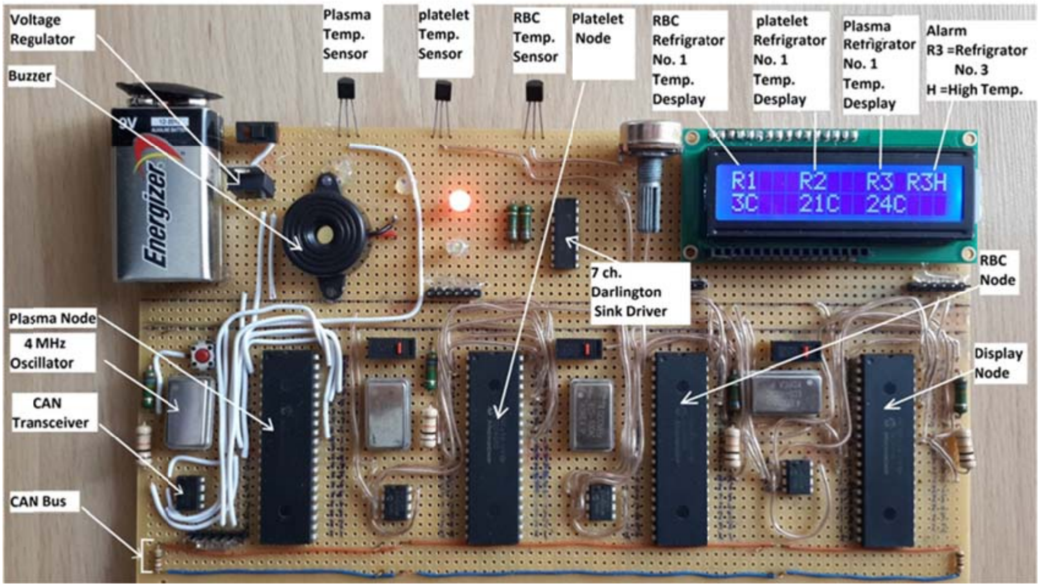

Fig. 10 The Overall Hardware System Board

VI. Conclusion

The four nodes CAN bus system designed and implemented successfully and the data exchanges between all the four nodes. The PIC18f458 is used in each of the four nodes. The PIC18f458 that used in display node interfaced with an LM35 temperature sensor and the CAN bus. The node is read the temperature from the sensor and build a CAN package data and then put it on the can bus as well as the node read the data on the bus successfully. Another node has been designed in this project that connected to an LCD and has sent the data to show on the LCD successfully. The LCD (display) node could send and read the data on the bus perfectly, Also it turned on and off the LED and Buzzer successfully in case the temperature is greater or less than the limited range. We conclude that one of the most important and sensitive parts in CAN system at the baud rate and timing parameters. It's necessary to follow the CAN data sheet and all the component data sheets, which has been used in the design, to build the correct timing parameters and consequently get the accurate interface between the PICs that have been used in the CAN bus system.

Fig. 11 Oscilloscope measuring data flow on the bus of CAN bus system

References

[1] CAN specification version 2.0 Robert Bosch GmbH, published by copper hill Technologies Corporation.

[2] Steve Corrigan, "Introduction to the Controller Area Network", Published by Texas Instruments Application Report, SLOA101A, August 2002–Revised July 2008.

[3] Texas Instruments, "Introduction to the Controller Area Network (CAN)," *Application report,* SLOA101A–August 2002–Revised July 2008.

[4] U. Hashim, M. N. Haron," Design of Digital Display System for ISFET pH Sensor by Using PIC Microcontroller Unit (MCU)"IEEE C.P.,pp.148-152,2009.

[5] Microchip technolog Inc, "High-Speed CAN Transceiver," *MCP2561/2, Device Document DS25167B,* 2013.
[6] Singh, S.A.J., Raviram, P., ShanthoshKumar, K., "Embedded Based Green House Monitoring System Using PIC Microcontroller'' IEEE ICGCCEE, 2014, pp.1-4.
[7] U. Hashim, M. N. Haron," Design of Digital Display System for ISFET pH Sensor by Using PIC Microcontroller Unit (MCU)"IEEE C.P.,pp.148-152,2009.
[8] National Semiconductor, "LM35 Precision Centigrade Temperature Sensors Data sheet", National Semiconductor Corporation, November 2000.
[9] Microchip Technology Inc, "28/40/44-Pin, Low-Power, High Performance Microcontrollers with XLP Technology," *PIC18(L)F2X/4XK22 Data Sheet*, ISBN: 9781620763131, USA 2012.
[10] Microchip Technology Inc, "Understanding Microchip's CAN Module Bit Timing," Document AN754, DS00754A, 2001.