Intelligent Sensor for Weight Measuring, Qualifying and Inspecting

Tsvetan Petkov Shoshkov, Stoyan Nikolov Jilov, Georgy Slavchev Mihov and Vasil Dimitrov Gunev

Abstract - **A solution of an intelligent sensor for weight inspecting is developed. It uses a specialized bridge transducer AD7730 for analog-to-digital conversion that controls two thensometric full-bridge sensors and transmits the result through a synchronous serial interface (SPI). In addition, a micro-controller unit (MCU) MC9S08QE8 is embedded in the controller module. It is used for organizing the operation of the intelligent sensor in a field local area network, using bus topology of standard RS485 and Modbus communication protocol. The intelligent sensor could be applied as a weight inspector for control and dosing the production**

Keywords – **weight measurement, industrial controller, Modbus LAN**

I. INTRODUCTION

The local computing network is the field for realizing a flexible automotive manufacture. Distributing the work between different devices in a network decreases the load of the main controller.

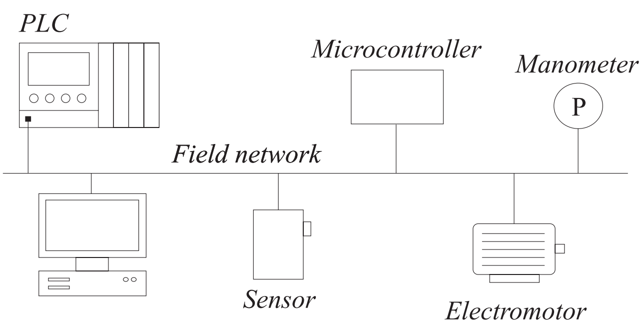

Fig. 1. Industrial Field Local Network

A main point of the manufacturing is the control of the production quantity. A basic method for its realizing is the measurement of the products weight. The weight measurement itself is relatively a routine activity [1]. The concrete circumstances, in which the weight measurement is done, make it difficult and set a lot of requirements for its realization [2, 3]. Such problem is the measurement in the industrial environment. There are a lot of significant interferences, which are an obstacle for the correct measurement.

An intelligent weight sensor is developed with embedded driver for communication under Modbus protocol. Using the network it can be configured and also it can deliver processed data for the measured weight. In Technical University – Sofia there is one controller for weight measuring [7]. The current development uses more present and more modern microcontroller. The communication protocol for the local network – Modbus [8], is wide spread and is relatively more simple and easier to use and realize. A disadvantage of the protocol is that it can have only one main device in the Modbus network. However, that is not obligatory and moreover it makes the network simpler.

II. ORGANIZATION OF THE CONTROLLER

A controller is developed, which is an intelligent sensor for weight measuring, qualifying and inspecting. The flow char of the controller is shown on Fig. 2.

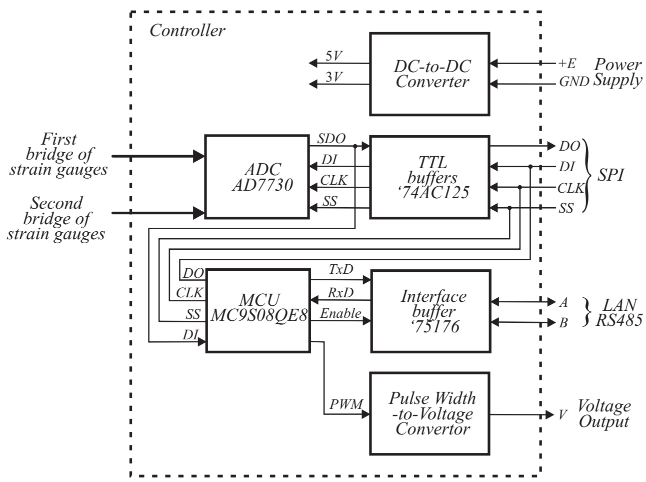

Fig. 2. Organization and flow char of the Controller

A specialized bridge transducer AD7730 of Analog Device Inc [4] is used for the analog to digital converter (ADC). It has two measurement channels, working with common reference voltage. The controller receivs the signals from the sensor and converts them in digital code, processes the data according to its configuration and transmits the results through a Serial Peripheral Interface (SPI). The principle scheme of the bridge transducer connections is shown on Fig. 3.

The serial interface is buffered using three-state buffers

Ts. Shoshkov is with the Faculty of Electronic Engineering and Technologies, Technical University of Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: bladejuven@gmail.com
St. Jilov is with SPV Ltd, Patriarh Evtimii blvd. 155, 6000 Stara Zagora, Bulgaria, e-mail: spv@abv.bg
G. Mihov is with the Department of Electronics, Faculty of Electronic Engineering and Technologies, Technical University of Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: gsm@tu-sofia.bg
V. Gunev is with SPV Ltd, Patriarh Evtimii blvd. 155, 6000 Stara Zagora, Bulgaria, e-mail: spv@abv.bg

74HCT125. Outputs SS, DI, SDO, CLK are connected to additional embedded microcontroller (MCU) MC9S08QE8 of the series of microcontrollers MC68HCS08 of Motorola/Freescale [5].

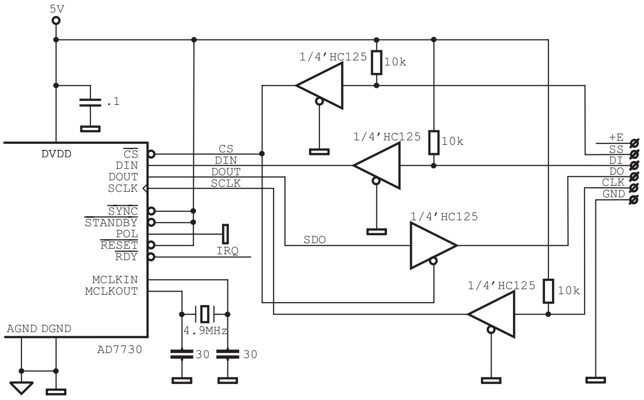

Fig. 3. Principle Scheme of the Digital Part of the Controller

The task of the microcontroller is to organize the work of the controller module and to make available the connection to a field local network using the bus topology of the physical standard RS485.

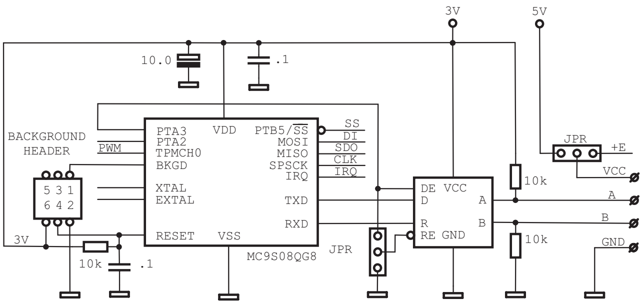

Fig. 4. Principle Scheme of the Local Network Organization

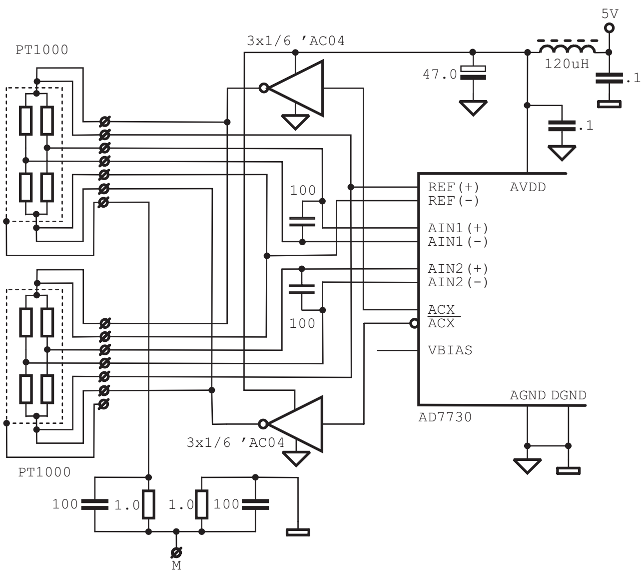

Fig. 5. Principle Scheme of the Analog Part

The connection through the local network is half duplex. The interface RS485 is an alternative to the SPI interface. The controller module can work either using SPI or using the RS485 interface, The SPI interface can be used by using outputs SS, DI, DO, CLK and GND, shown on Fig. 2 and Fig 3. The principle scheme of the local network organization is shown on Fig. 4.

Outputs A, B and GND are the signals that are connected to the local network.

The weight measurement is done using sensors of thensometric type, connected in a full bridge scheme – strain gauge bridges. The principle scheme of the connecting of the strain gauge bridges is shown on Fig. 5.

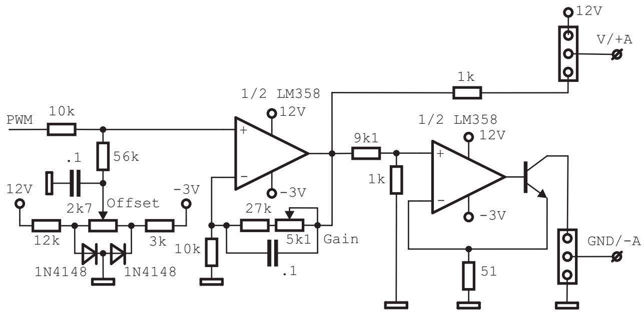

Fig. 6. Principle Scheme of the Pulse Width-to-Analog Converter.

The smart sensor is provided with analog output, generating current or voltage signal, proportional to the measured weight. The digital-to-analog conversion is performed by the one of embedded timer of the microcontroller. It produces a frequency with proportional to the digital code duty cycle. The shown in Fig. 6 scheme converts the pulse width of the timer into an analog signal.

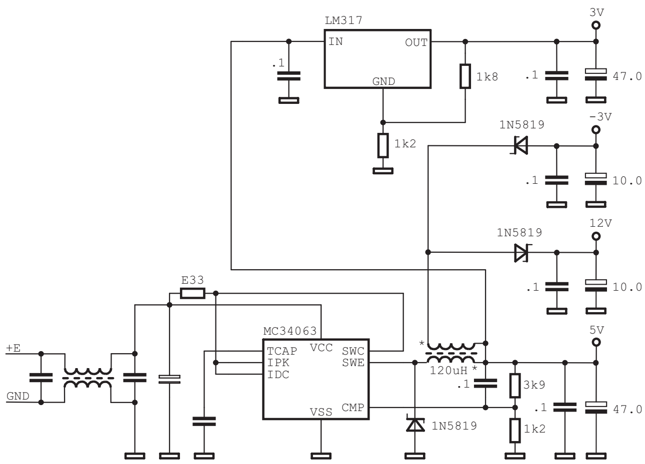

Fig. 7. Power Supply Voltages

For the proper functioning of the device it is needed to be developed power supply voltages. They must satisfy the needs of all modules. It is provided a step-down DC-DC converter. It uses a pulse width modulation scheme MC34063 that controls the main switch of the step-down converter. The switch control is done depending on the input voltage level. That allows the scheme to be able to work using high voltage input with wide range from 12V to 24V. Fig. 7 shows the scheme that produces digital power supply voltages 5V and 3V, needed by the modules in the digital part of the scheme, and an analog power supply voltage 5V, needed by the modules in the analog part of the scheme.

III. Computer Software of the Controller

The work of the controller module is divided in two modes – configuration mode, in which the analog-to-digital converter is calibrated, and work mode, in which the actual object weight measurements are processed. The work of the controller is realized by using a real time operational system. It has a dispatcher that switches the current tasks. It has two interrupt ports – from system timers and from the peripheral modules. On Fig. 8 it is shown the structure of the organization of the controller work.

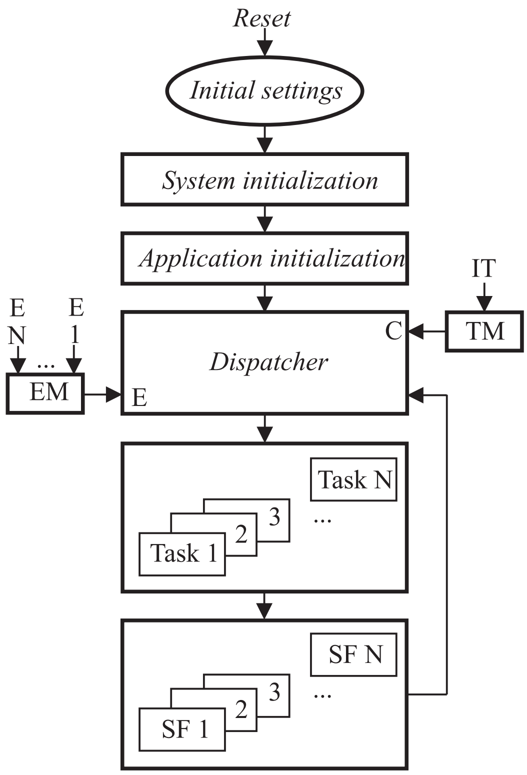

Fig. 8. Work Organization of the Controller

After starting up the system, there are made initial setting and system and application initializations are performed. After that the dispatcher takes control on the switching between that tasks and system functions. The tasks are processing the measured data received and servicing the communication over the local network.

IV. Communication Driver under Modbus Protocol

The controller has built in communication driver for local network under Modbus protocol. It has the following input points: interrupt of the system timer, interrupt of the serial channel, initialization of the network, start of receiving, start of transmitting [8].

Modbus is a communication protocol, developed by Modicon systems. In the standard Modbus network there is one master and up to 247 slave devices. Each of them has a unique slave address from 1 to 247. Only the master device has the right to send requests over the network. The slave devices are scanning the network for requests addressed to them, process the corresponding instructions, and after that sends a reply to the master device.

The Modbus connection allows two basic modes of transmitting – ASCII (American Standard Code for Information Interchange) and RTU (Remote Terminal Unit). The Modbus/ASCII messages uses ASCII format of information. The Modbus/RTU format uses binary coding. All subscribers of a Modbus network must use the same mode of data transmitting.

Each Modbus/ASCII message begins with the transmission of a symbol ‘:’ (ASCII code 0x3A) and ends with symbols CR (Carriage Return) and LF (Line Feed) – see Fig. 9.

‘:’	*Slave address*	*Functional code*	*Data*	*LRC*	*CRLF*
1 byte	2 bytes	2 bytes	N bytes	2 bytes	2 bytes

Fig. 9. Message Structure in Modbus/ASCII mode

On Fig. 10. it is shown the structure of a RTU message. The transmission of each message in Modbus/RTU mode is performed before an empty time interval, that is longer than the needed for 3.5 chars transmitting time interval. If the receiver detects a ‘gap’ more than 1.5 chars, it is considered that a new message is arriving and the receiving buffer is empty.

Slave address	*Functional code*	*Data*	*CRC*
1 byte	1 byte	0 to 252 bytes	2 bytes

Fig. 10. Message Structure in Modbus/RTU mode

The first element of each message is the address of the receiver. This parameter consists of one byte information. The Modbus/ASCII address is coded with two hexadecimal chars. The Modbus/RTU address is only one byte.

The second byte of the message is the function code. It defines the task that will be launched by the Slave controller. When a Modbus receiver ‘responds’, it uses the same function code as in the request. When an error occurs, the master bit in the function code is set to one. By this way the Master controller is able to make the difference between the correct and the wrong answer.

As a result of a received request there might be four followed different cases:

– The request is successfully received by the Slave controller and there is a valid response.

– The request is not received by the Slave controller and there is no response.

– The request is received by the Slave device with parity, CRC or LRC error. The Slave device ignores the request and doesn’t send a response.

– The request is received without any errors but it cannot be launched by the Slave device because of another reason. The master bit of the function code in the response is set to one.

The Modbus standard uses preservatives for error detecting. While transmitting each byte of the message an odd/even check is made. The last block of each message is a CRC or LRC.

The built in driver for communication uses both ASCII and RTU mode of the Modbus communication protocol.

The computer program of the system has a command interpreter. The command interpreter work structure and its connection to other modules of the computer program is shown on Fig. 11.

The received command determines what operations to be done by the device and what will be the content of the answer. The command is integrated in the communication protocol as a first byte in the “Data” field of the message. By transmitting a command over the local network it can be processed a calibration of the ADC module or can be set a start of measured data transmission. The command interpreter supports the following commands:

– Inquiry for presence
– Change of the controller address
– Loading all parameters of the ADC
– Loading the work mode of the ADC
– Loading the averaging number of measured samples
– Loading the work mode of the filter
– Loading the ADC offset
– Loading the calibrating offset
– Loading the calibrating gain
– Reading all parameters of ADC
– Reading the work mode of the ADC
– Reading the averaging number of measured samples
– Reading the work mode of the filter
– Reading the ADC offset
– Reading the calibrating offset
– Reading the calibrating gain
– Receiving weight measured data from the controller
– Receiving data from the first control channel
– Receiving data from the second controller channel

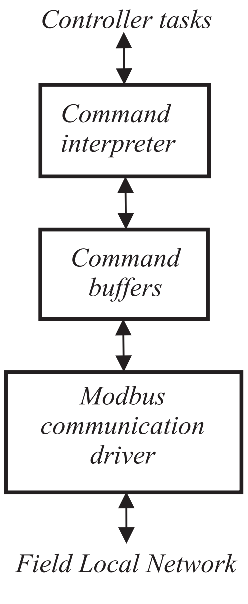

Fig. 11. Command Interpreter

As a main station it is used a personal computer (IBM PC). There is a terminal program for PC developed in Technical University – Sofia. It can be used for communication in a local network. It has a built in driver for communication. The communication driver is processed and adjusted to work under the Modbus communication protocol. The connection between the personal computer and the developed intelligent sensor in a local network gives the opportunity for testing the work of the device. Configuration and work in real conditions can be done.

V. Conclusion

It is developed a controller that is an intelligent sensor for weight inspection. For the analog-to-digital conversion is used the bridge transducer AD7730. The signal module controls two strain gauge bridges, converts the data in digital form, process it according to the corresponding configuration and transmit the results using a SPI interface. There is additionally built in microcontroller MC9S08QE8. It organizes the work of the signal module in a field local network using the topology of the physical standard RS485 and Modbus communication protocol. The developed control module can be configured over the network. The computer program of the microcontroller also has a built in command interpreter. It interprets the received command and determines the tasks that must be processed by the device. As a main station it is used a personal computer, that also has a built in driver for communication under the Modbus protocol.

There is another system for weight measurement developed in Technical University – Sofia. However, the current development uses more modern microcontroller and the communication protocol is more simple and easy to be realized and used.

The intelligent sensor measures the data, processes it and makes it accessible to the main station over the local network. This decreases the work load of the main device.

Acknowledgement

This research was supported by the NIS TUS funded project, No 091ni040-03/2009.

References

[1] K. Hoffman. *An Introduction to Measurements using Strain Gages*, HBM GmbH, Darmstadt, 1989.
[2] S. Jilov, V. Gunev, R. Ivanov, G. Mihov, N. Miteva, N. Ivanova. *Microprocessor system for controlling of a single or a group of machines with discrete work*. National science-technical conference “System and Machines for Food industry”, St. Zagora, microf. № 2-9, page 14. 1983.
[3] I. Dimitrov, G. Mihov. *Program and Apparatus Realization of a Controller for Discrete Production*, ELECTRONICS ET2000, b. 2, Sozopol, p.3-9. 2000.
[4] www.analog.com/en/prod/0,2877,AD7730,00.html .
[5] Freescale. *HCS08 Microcontrollers*, MC9S08QE8 Rev. 1.01 10/2005.
[6] Measurements Group. *Errors due to Transverse Sensitivity in Strain Gages*, Experimental Techniques No.1 p.30-35, 1983.
[7] G. Mihov, E. Dimitrov, V. Gunev. *Industrial Controller for Weight Measurement*, The International Scientific Conference EIST 2001. vol. III, Bitola, Macedonia, p.649-654, 2001.
[8] Ts. Shoshkov, G. Mihov, V. Manoev - “Small Local Network for Microprocessor Controllers” – Electronics’ 2008 Sozopol, Volume 1, page 93.
[9] St. Jilov, V. Gunev, G. Mihov. *Weigh inspector for quantitative and qualitative production control,* International scientific conference “Economics and Society Development on the Base of the Knowledge”, Stara Zagora, CD, 2009.