7th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
Suceava, Romania, May 27 – 29, 2004

PROTOCOL FOR COMMUNICATION BETWEEN TELEMETRY SYSTEM AND SENSORS IN BOREHOLE MEASUREMENT INSTRUMENTS

Miloš SLANKAMENAC [1], Krešimir KNAPP [2], Miloš ŽIVANOV [1]
[1)] *University of Novi Sad, Faculty of Technical Sciences, 21000 Novi Sad, Serbia and Montenegro, miloss@uns.ns.ac.yu*
[2)] *Hotwell Ges. m. b. H. Oedenburger Strasse 6, 7013 Klingenbach, Austria*

Abstract. *In this paper are presented an implementation and algorithm for communication protocol between Telemetry system and sensors in borehole measurement instruments. This protocol is called SIPLOS (Simultaneous Production Logging String) and it is used in Hotwell company as part of larger system for borehole investigations. Protocol between Telemetry tool and Calliper-Fullbore Flowmeter tool is described in details.*
Keywords: *protocol, Calliper-Fullbore Flowmeter tool, Telemetry tool, measurement, sensors .*

Introduction

The SIPLOS (Simultaneous Production Logging String) system consists of the Surface unit, Telemetry tool and other tools in a logging string. The Telemetry tool is the first and a necessary tool in a string of production tools and is connected on the top of the cablehead. It has capability for logging of CCL, internal temperature, external temperature, pressure, fluid identifier and gamma ray. Also, there are all necessary electronics (amplifiers, signal conditioners, DC/DC converter and line driver) inside the Telemetry tool. In the bottom of the Telemetry tool there is a single connector with line voltage where it is possible to connect other tools in a string for simultaneous production logging.

Calliper-Fullbore Flowmeter tool (CFF) combines a two axis (X-Y) calliper with a fullbore flowmeter in a single device. X-Y calliper is designed for downhole pipes diameter measurements in two axes, X-Y. Measuring range of pipe diameter is $4^{1/2}$"-7". The pipe diameter that is sensed by the calliper arms is magnetically transferred to a linear position sensor. The position sensor is a part of an electronics that converts linear position to the output pulses.

Flowmeter performs bi-directional measurement of the rotational speed and direction of a turbine wheel (spinner or impeller), in relation to fluid flow inside the well. The four-arm device is designed to minimize the risk of jamming inside pockets and other well fittings, and uniquely allows the spinner to rotate whilst partially closed. This enables a range of tubing/casing to be logged using only a single cage assembly, thereby minimizing inventory and capital cost. Figure 1 depicts connection between Telemetry Tool and CFF.

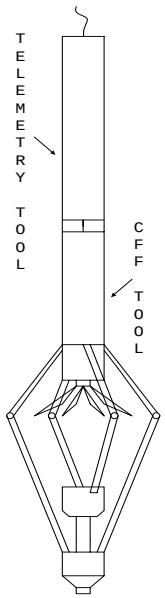


Fig. 1. Connection between Telemetry Tool and CFF

The goal of this project is to provide an accurate and reliable communication between CFF tool with Telemetry tool and the Surface unit in very difficult conditions for measurement in

borehole. The main problem is high and variable temperature (up to 180 ºC) causing significant drop of frequency of oscillator in microcontroller.

Transmission of data from CFF sensors to CFF protocol device

The calliper arm position is measured by using a transformer based position sensor. A metal rod is moved within the transformer core that has the effect of varying the coupling between transformer's primary and secondary. The transformer primary is driven by a fixed amplitude voltage pulse and the degree of coupling is determined by measuring the resultant signal in the transformer secondary. Thus the amplitude of the detected secondary signal is proportional to the arm position and hence to the diameter of the well-bore. The complete calliper system is controlled by a microcontroller PIC 16F876. It controls timing of the primary drive signals and uses an internal A-D converter to sample the transformer secondary signals. In operation, voltage pulses are applied to the transformer primary. The voltage across the transformer secondary will rise sharply with the rising edge of the pulse and then decay at a rate dependent on the inductance of the transformer system. This inductance depends on the position of the sensor rod within the transformer core. If the decay waveform is sampled at a precise time after the rising edge of the primary pulse, the magnitude of the sampled voltage will be linear proportional to the rod position. This voltage is measured by the A-D converter. The clock rate for microcontroller is set at 16.384 MHz in order to achieve the correct comms baud rate of 115.2 kbaud. The processed data are transmitted by UART to the CFF protocol device in standard RS232 format with 1 start bit, 8 data bits and 1 stop bit (no parity).
Figure 2 represents a CFF calliper block diagram.

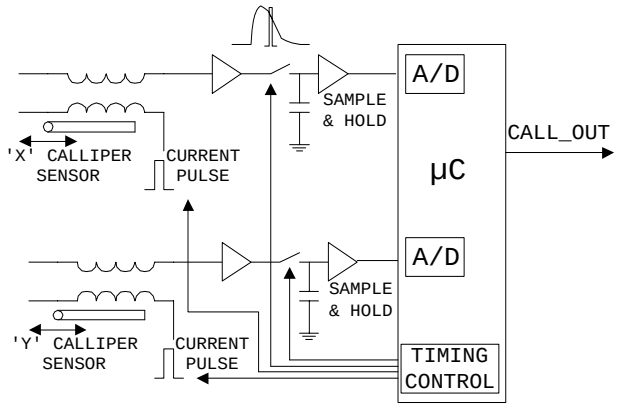


Fig. 2. CFF calliper block diagram

The flowmeter uses an array of 5 fixed Hall effect sensors operated by a rotating magnet assembly containing two outward facing poles. This results in 10 sensor operations per revolution of the impeller. The sensors are scanned at a high sample rate by a microcontroller. The sequence of sensor operations indicates the direction of rotation. The rate of sensor operations is proportional to the flow rate. The flow rate and direction signals are input to the CFF protocol device as logic voltage levels. The microcontroller is able to detect the absence of a sensor signal due to failure and can compensate it so that the operator will see no change in flow rate on a log, only a reduction in resolution. A fault indication signal is available from the microcontroller to aid in servicing and fault finding.
Figure 3 represents a CFF flowmeter block diagram.

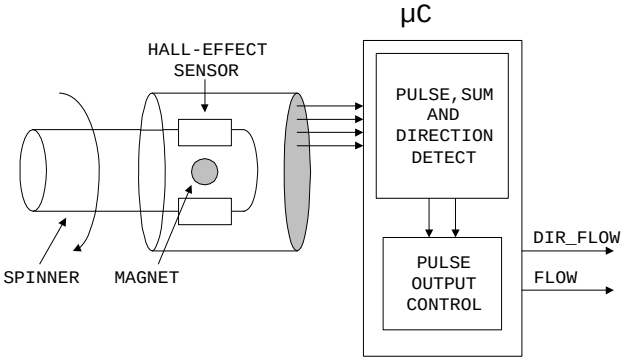


Fig. 3. CFF flowmeter block diagram

Data flow of SIPLOS protocol

The tool string sends a package of all sensors data to the outside measurement system in each 200 ms. Block of 200ms is divided into two parts:

- first part is a pause that lasts 16.667 ms (50ms/3), and for that time there is no voltage change on LINE.
- second part that lasts 183.333 ms (550ms/3) during which data are sent (there are negative pulses on LINE).

Data are sent in blocks of 8-bit words and there are 20 words in each package. An 8-bit word that lasts 9.167 ms (55ms/6) consists of a start bit, 8 data bits and two stop bits. Each bit lasts 833.333 μs (5ms/6). It can be said that the whole package of 200 ms that lasts 240 bits is divided into two parts: a pause of 20 bits and 20 words of data that last 220 bits. This can be seen in the following figures:

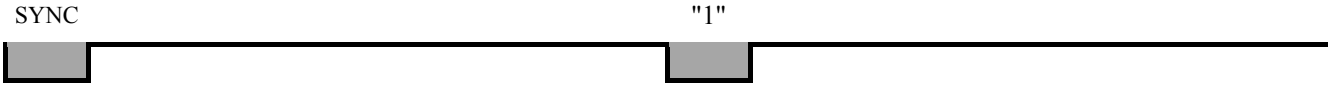
PAUSE	1st word	2nd word	3rd word	4th word	5th word	6th word	7th word		14th word	15th word	16th word	17th word	18th word	19th word	20th word
20bits=50ms/3 16.667ms	220bits=(55ms/6)*20 183.333ms														
200 ms															

Fig. 4. One whole package of data

START BIT	1st BIT (LSB)	2nd BIT	3rd BIT	4th BIT	5th BIT	6th BIT	7th BIT	8th BIT (MSB)	STOP 1 BIT	STOP 2 BIT
5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6	5ms/6
833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs	833.333μs
(5ms/6)*11=55ms/6										
9.167 ms										

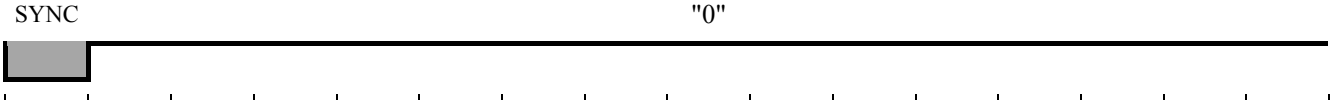
Fig. 5. One data word

Start bit and data bits are represented on LINE with negative pulses. Each of them has a SYNC pulse that lasts 52.083 μs (1/16 of the whole bit, exactly: 5ms/96) at the beginning of the bit. If DATA BIT=1 (logical HIGH) there is also the same type of negative pulse in the middle of the bit. START BIT is presented the same as DATA BIT=1. If DATA BIT=0 (logical LOW) there is no negative pulse in the middle of the bit. 2 STOP BITS are presented as pause that lasts 2*833.333 μs and during that time there are no pulses on LINE. The first data bit after START BIT is a LSB, the eight bit is a MSB. This is presented in the following figures.



52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs
5ms/12=416.667μs								5ms/12=416.667μs							
5ms/6=833.333μs															

Fig. 6. DATA BIT=1



52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs	52μs
5ms/12=416.667μs								5ms/12=416.667μs							
5ms/6=833.333μs															

Fig. 7. DATA BIT=0

Figure 8 shows signal on LINE when decade number 43275 is sent as 16-bit binary data. Decade number 43275 equals binary number 1010100100001011, but sending order on LINE is reversed: 1101000010010101.

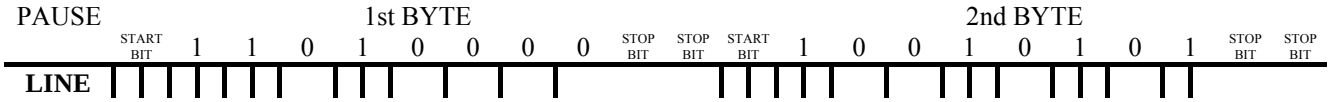


Fig. 8. Sending decade number 43275

START BIT is presented as DATA BIT=1, and 2 STOP BITS as a pause that lasts 2 bits, so it is possible to convert pulses from LINE into standard RS232 signal with BAUD RATE=1200 (1 START BIT + 8 DATA BIT + 2 STOP BIT). It can be seen that after a long pause that lasts 16.667ms there is a START BIT of the first word (SYNC pulse on LINE). but because the last word of previous package finished with two STOP BITS there is no voltage change on LINE for the time: 16.667 ms (pause) + 1.667 ms (2*STOP BIT) + 7(or 15)*(833.333μs/16) = 18.722 ms (or 19.114 ms).The last number in this expression depends on the value of the last data bit of previous block of data (for "0" is longer then for "1").

All time periods are generated from the same clock oscillator in the tool, so any change in frequency of system clock generates linear changes in all time periods. For example, if frequency of clock is lower for 10% than duration of pulses is 57.2 μs (52+5.2) μs, a whole package lasts 220 ms etc. Surface electronics and software make necessary corrections in order to get correct data.

Data flow in Telemetry tool

The Telemetry Tool takes for all its sensors ten 8-bit words so for running other tools there are still free ten 8-bit words. After long pause the first 8-bit word is reserved for internal temperature data. The second and third 8-bit words are reserved for CCL. The fourth and fifth words are reserved for external temperature data. The sixth and seventh words are reserved for pressure data. The eighth and ninth words are reserved for fluid identifier data. The tenth word is reserved for gamma ray data. All this is presented in the following figure.

The Telemetry Tool sends:

- all data from its six sensors to LINE
- START BITS for all twenty words to LINE
- SYNC PULSES of all DATA BITS for all twenty words to LINE

If a sensor is not present in string or if there is a failure in its position then all data would be zeros.

1st word	2nd word	3rd word	4th word	5th word	6th word	7th word	8th word
LSB** Select	15-bit **CCL/T-int**		16-bit **T-ext**		16-bit **pressure**	8-bit **FI**	8-bit **GR**

Fig. 9. One Whole Package of Data from the Telemetry Tool

Data flow in CFF tool

Figure 10 depicts block diagram of hardware realization of CFF tool part, for communication between sensors and surface system, that is called CFF protocol device. In this project are chosen PIC 16F876 as μC_1 and PIC 14000 as μC_2, because tested on higher temperature they worked very reliably [1].

LINE is bi-directional signal from Telemetry tool that consists of a line voltage (70V) and negative pulses (START, STOP, SYNC and DATA bits). Line voltage is transmitted by surface unit to all tools. Telemetry tool transmits START, STOP and SYNC pulses to all other tools. DATA bits are transmitted by all measuring tools to surface unit in determined

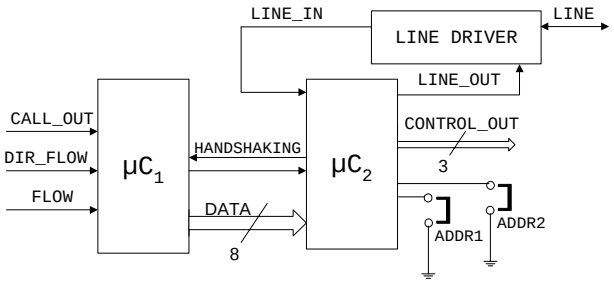


Fig. 10. Block diagram of CFF protocol device

interval of time. LINE DRIVER is a device that converts line voltage into CMOS voltage level, discriminates pulses from LINE signal and makes LINE_IN signal. LINE_IN signal contains START, STOP and SYNC pulses that determine timing for sending DATA from CFF to surface unit. LINE DRIVER receives LINE_OUT signal from μC_2 that is converted into corresponding pulses, positioned in the middle of a data-bit frame, whose duration is 50μs ± 10% and sends it to surface unit. CONTROL_OUT are signals for verification purpose which contain an error signal and two signals for synchronization verification. μC_1 receives signals CALL_OUT, DIR_FLOW and FLOW from sensors. CALL_OUT contains information about pipe diameter from calliper in RS232 protocol form [2]. DIR_FLOW and FLOW contain information about flowmeter impeller direction and frequency given in a pulse form. Algorithms for programming both microcontrollers have to work reliably for both high and variable temperature conditions. Oscillator's frequency (internal or RC) can be decreased even for 30%. Fig. 11 shows algorithm for software in μC_1. There is one main part, after the initialization and catching sinhronization pause, which is always repeated. RECEIVING DATA means receiving two bytes data from CALL_OUT signal, whilst counting of pulses from FLOW signal is performed continually during the entire algorithm. When impeller is changed a direction, a new direction will be indicated and counting pulses in previous direction will be summed. When request for sending DATA is received from μC_2, all data is summed and transferred to the second microcontroller every 200 ms.

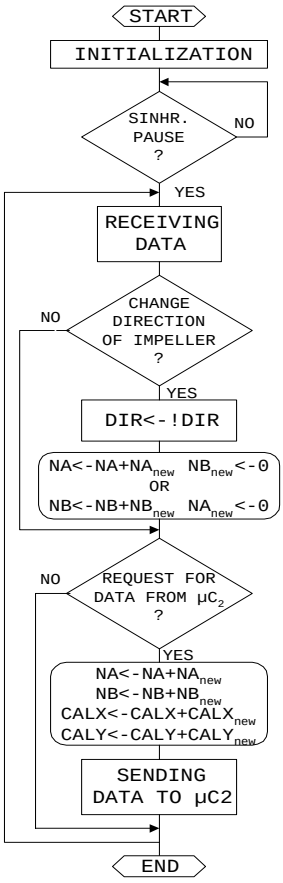


Fig. 11. Algorithm for μC1

Figure 12 shows algorithm for software in μC2. After initialization program waits a sinhronization pause that last 16,667 ms. Because of varying internal frequency of μC2 microcontroller clock, it's not reliable to calculate time intervals based on it. Calculating of time intervals has to be performed based on measuring timings on LINE_IN signal. In INITIAL MODE measuring timings, counting number of pulse and calculating initial parameters are performed during 200 ms. Initial parameters involve time intervals when START, STOP and DATA bits appeared on LINE. If there are too many or few pulses or their duration isn't proper, the microcontroller μC_2 is restarted by Watchdog Timer. In the FIRST MODE the measuring duration of 12 bytes and calculating parameters without modulation (sending DATA according to SIPLOS protocol) is performed. In the NORMAL MODE the address where data should be transmitted via LINE_OUT signal is determined by position of jumpers ADDR1 and ADDR2 (Fig. 10). There are two possibilities where data can be

modulated in a package of data: to addresses 11 to 18 or 13 to 20. After that μC_2 requests and receives data from μC_1. Modulation is performed according to SIPLOS protocol based on measured time intervals.

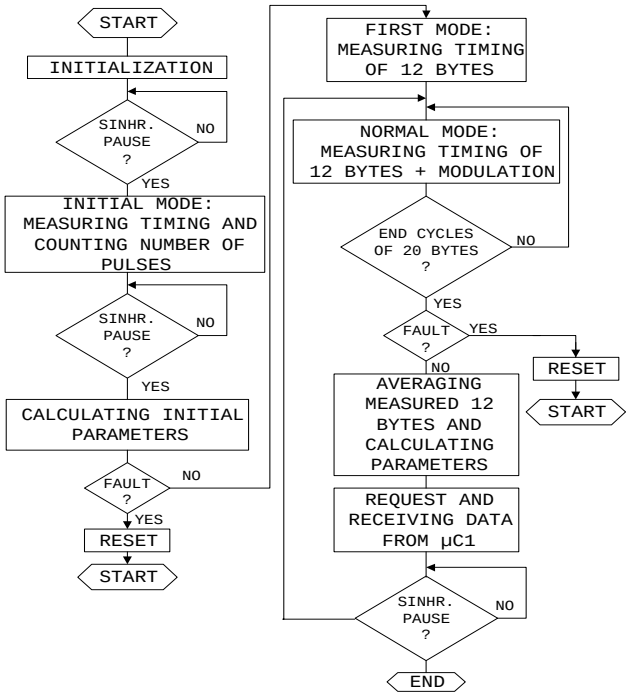


Fig. 12. Algorithm for µC2

Simulation results

Simulations of work of microcontrollers PIC 16F876 and PIC 14000 were performed successively by MPLAB IDE [3]. Simulator Stimulus options were used to simulate input signals. Both of microcontrollers were simulated separately. DIR_FLOW, ADDR1, ADDR2 and HANDSHAKING signals were simulated by Asynchronous Stimulus option where it is possible to change state of particular pin during the simulation. CALL_OUT and LINE_IN signals were simulated by Pin Stimulus option. CALL_OUT was simulated by setting the duration of sequence '0' and '1' in RS232 form for 115,2 kbaud rate protocol. LINE_IN was simulated by setting the duration of sequence '0' and '1' in SIPLOS protocol form. FLOW signal was simulated by Clock Stimulus option, setting the constant period of sequences '0' and '1'. Output signals, SFR registers and memory content were observed during the simulation step by step [3]. All of input signals were simulated for ideal and the worst temperature conditions [1] and expected output signals were obtained.

Experimental results

CFF protocol device was tested by using appropriate input signals from serial and parallel port of PC whilst output signals were observed by oscilloscope. These input and output signals were similar to those obtained by software simulation. This project hasn't been finished yet. When some mechanic parts of CFF tool are finished, the whole string of tools should be tested in borehole conditions.

Conclusion

SIPLOS protocol is used in digital system for borehole investigations. Digital systems measure more parameters at the same time than analogue systems. Because of this advantage, process of logging is much shorter and cheaper. Digital logging string is smaller, reliable and more effective than analogue logging string. Based on software simulation and experimental results we can conclude that CFF protocol device satisfies SIPLOS protocol and works properly.

References

[1] Bilas, V., Vasić, D., Ambruš D. (2003) *SIPLOS Flowmeter – technical documentation* Hotwell Ges.m.b.H, Austria

[2] Matić, N. (2002) *PIC mikrokontroleri* Beograd: MikroElektronika

[3] Microchip Technology Inc. (2001) *MPLAB IDE User's Guide*

Acknowledgement

Many thanks to Hotwell Ges. m. b. H. Austria for supporting this work.