10th INTERNATIONAL SCIENTIFIC CONFERENCE
ON DEFENSIVE TECHNOLOGIES
OTEH 2022
Belgrade, Serbia, 13 – 14 October 2022

# DEVELOPMENT OF THE DEVICE BASED ON THE DSPIC30F6014A MICROCONTROLLER FOR MEASUREMENT OF AXLE POSITION ANGLE

VLADIMIR LAPČEVIĆ
Military Technical Institute, Belgrade, vladimirlapcevic@mts.rs

**Abstract:** *Development of the device for measurement of axle position angle is presented in this paper. In many cases where the motors are used there is a need for measurement of axle position angle. This document presents the detailed development of hardware and software which provide precise measurement of axle position angle. Created device is based on the microcontroller dsPIC30F6014A. Transducer of angle which is used is absolute encoder.*

**Keywords:** *microcontroller dsPIC30F6014A, encoder, hardware, software .*

## 1. INTRODUCTION

Electric motors are used in servo controlling very often. In that case working body on the motor's axle is settled and it is necessary to be positioned very precisely. Sensor of angle has to precisely set working body in the determinated position. The encoder is a sensor of angle which gives information of the position of working body. Electronic device is connected to an encoder and takes values from the encoder and sends data to the computer. This paper presents the development of the device which takes serial data from encoder, then it does digital processing of data, where serial bits are packed into the digital word, and then this device sends digital word to computer by serial interface RS232. Communication between the device and the encoder is realised by RS485 serial interface. Complete electronics of the device is presented in this paper. Microcontroller dsPIC30F6014A [1] is used inside the device. Program for microcontroller dsPIC30F6014A [2] is written in C program language. The size of written program for microcontroller is 4 KB. Application program, which is installed on the computer and which enables communication with the device is written in C# program language. Development board of device is shown in the figure 1.

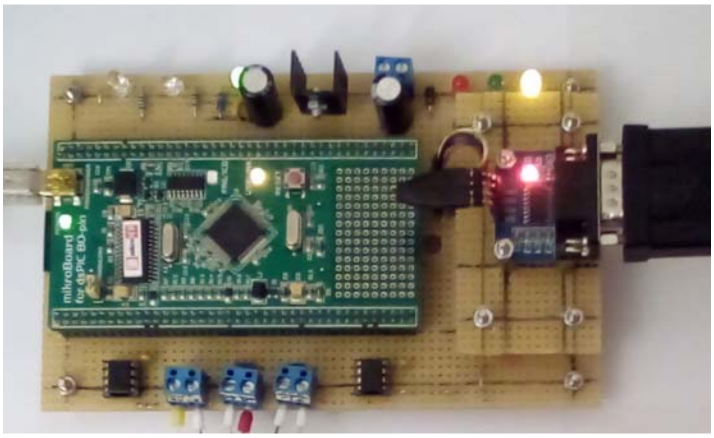

**Figure 1.** Development board of the device

## 2. ELECTRONICS OF DEVICE

### 2.1. Microcontroller dsPIC30F6014A

A microcontroller dsPIC30F6014A is the microcontroller which belongs to the family of digital signal controllers [3]. A digital signal controller is the microcontroller which is much better than the ordinary microcontroller because it has a digital signal processor (DSP) besides the microcontroller. The digital signal processor provides computing complex mathematical operations for a very short time: multiplying two 16-bits numbers, shifting 32-bits number and adding two 32-bits numbers. The digital signal processor enables multiplying two 16-bits numbers in one instruction's cycle. The digital signal controller is used in the real time system because of its speed. The digital signal controller reads signal from input sensors, processes all input signals and then generates control signals on its output. The digital signal controller must do all these operations in the defined time because only in that case a system is a real time system. The microcontroller dsPIC30F6014A and its connections to the rest of electronics inside the device is shown in the figure 2.

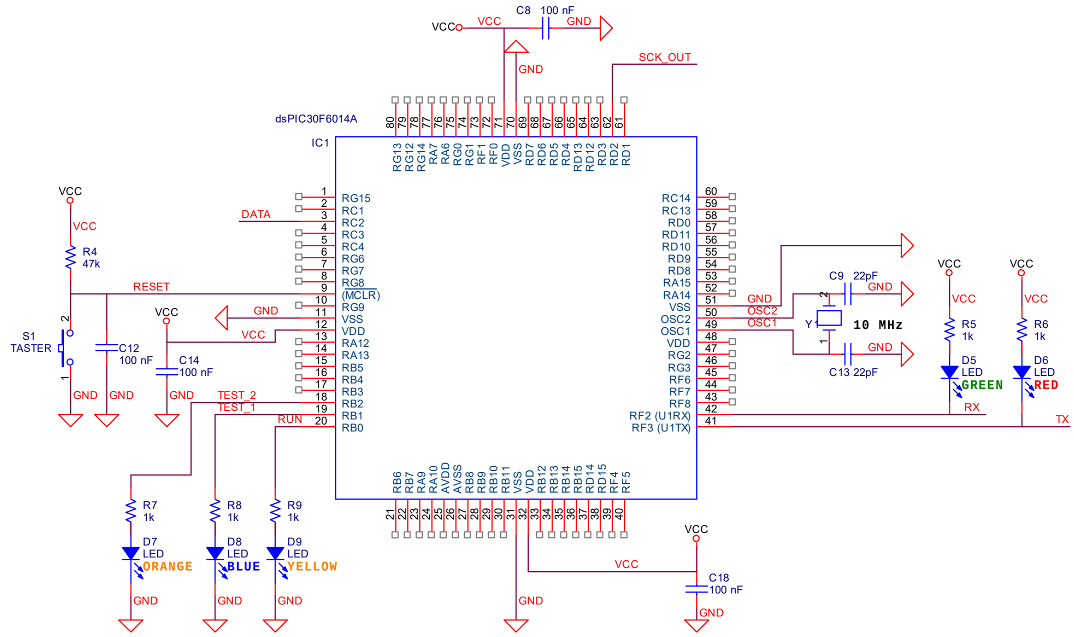


**Figure 2.** Microcontroller dsPIC30F6014A and its connections to the rest of electronics in the device

PLL oscillator is integrated into the microcontroller dsPIC30F6014A. It is possible to change the frequency of a microcontroller in reference of crystal quartz frequency, which is connected to the microcontroller, by this PLL oscillator. Frequency, on which microcontroller dsPIC30F6014A works, can be computed by next equation:

$$F_{CY} = \frac{K_P \cdot f_Q}{4} \qquad (1)$$

where is:

$f_Q$ – frequency of crystal quartz

$K_P$ – coefficient of PLL oscillator (4, 8, 16)

Coefficient of PLL oscillator has one of the next values: 4, 8 or 16. The value of this coefficient is asserted during setting configuration bits of a microcontroller. It can be seen from equation (1) that choosing value of coefficient $K_P$ can achieve that frequency of microcontroller which can be equal or higher in reference of crystal quartz frequency. In this case it is chosen that the value of coefficient $K_P$ = 8. In that way the results is two times higher frequency of the microcontroller than the frequency of a crystal quartz. As the frequency of crystal quartz $f_Q$ = 10 MHz, then the frequency of a microcontroller $F_{CY}$ = 20 MHz. The microcontroller dsPIC30F6014A sends clock to encoder and gets serial digital data from it. Communication between the microcontroller and the encoder is realized by serial RS485 interface. The maximum value of pulse and pause for this encoder is 12.4 µs. A clock, which the microcontroller sends to the encoder, has equal values of pulse and pause and they have the value 10 µs. Timer T1, which is integrated into microcontroller dsPIC30F6014A, is used for realization of the clock. Since pulse and pause of the clock are equal and have value of 10 µs, it is necessary that interruption of the routine of a timer occurs every 10 µs. Period of the interrupt routine occurrence is computed by next equation:

$$T_{INT} = n \cdot T_{CLK} \qquad (2)$$

Where is:

$n$ – number of ticks of timer until interruption occurs

$T_{CLK}$ – period of a timer's clock

Number n is the difference between the value of a register PR1 and the starting value of a timer TMR1. The starting value of a timer TMR1 is asserted to the zero. The timer counts ticks. The interruption occurs when the value of a timer is equal with the value of a register PR1. Period of timer's clock is equal to the reciprocal value of frequency of timer's clock:

$$T_{CLK} = \frac{1}{f_{CLK}} \qquad (3)$$

where is:

$f_{CLK}$ – frequency of timer's clock

The value of the prescaler $K_S$ of the timer T1 is asserted in the control register T1CON. Fourth and fifth bit in the control register T1CON are dedicated for setting the value of the prescaler $K_S$. These bits are called TCKPS bits. The value of the prescaler $K_S$ can be: 1, 8, 64 and 256. In table 1 values of TCKPS bits and values of the prescaler are shown.

**Table 1.** Values of TCKPS bits and values of the prescaler

| TCKPS<1:0> | $K_S$ |
|---|---|
| 00 | 1 |
| 01 | 8 |
| 10 | 64 |
| 11 | 256 |

Frequency of timer's clock is got as a divider of frequency of microcontroller $F_{CY}$ and the value of prescaler $K_S$ of that timer.

$$f_{CLK} = \frac{F_{CY}}{K_S} \quad (4)$$

Where is:

$F_{CY}$ – frequency of the microcontroller

$K_S$ – value of the prescaler of a timer T1 (1, 8, 64, 256)

Expression for time period repeating interrupt routine is based to equations (2), (3) and (4):

$$T_{INT} = \frac{n \cdot K_S}{F_{CY}} \quad (5)$$

The frequency of the timer's clock $f_{CLK}$ is equal to the frequency of the microcontroller $F_{CY}$ if the value of the prescaler $K_S$ is asserted to 1 in the control register T1CON. As a result, last expression can be written:

$$T_{INT} = \frac{n}{F_{CY}} \quad (6)$$

The variable n must have the value 200 because the frequency of the microcontroller $F_{CY}$ = 20 MHz and the interruption routine of a timer occur every 10 µs. It can be realized if the value of the register PR1 is 200 and the starting value of the timer T1 is asserted to zero every time when entrance to interruption routine occurs.

## 2.2. Serial interface RS485 through encoder

The encoder, which is used, has integrated RS485 interface. The signals from the encoder are differential signals. The connection between the microcontroller and the encoder is realized by RS485 interface. In the figure 3 serial interface RS485 is shown.

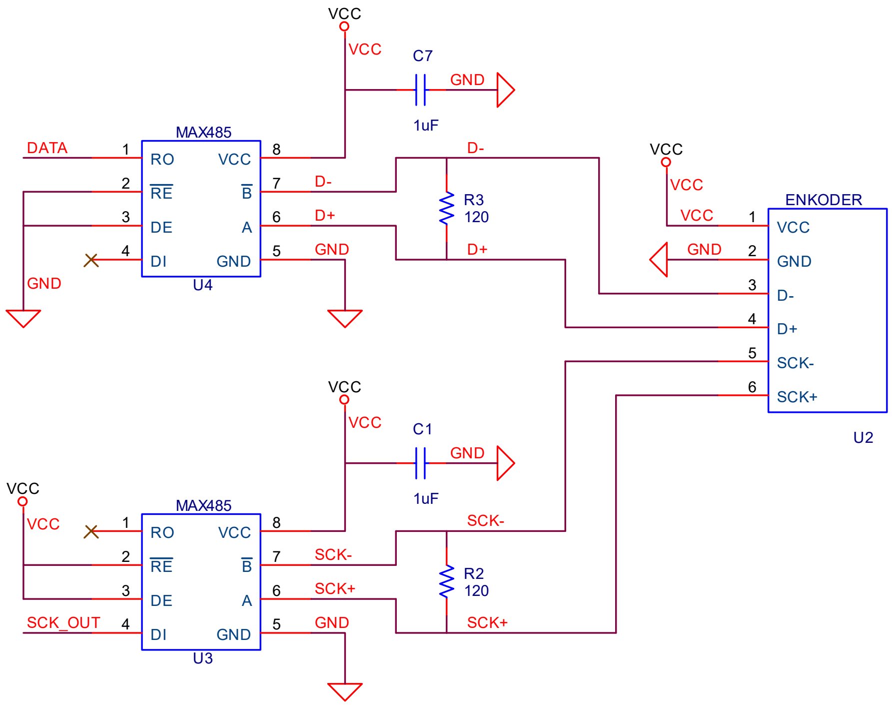


**Figure 3.** The serial interface RS485 between the microcontroller and the encoder

The integrated circuit MAX485 [4] is used for RS485 serial interface in this device. One integrated circuit MAX485 is used for transferring the signal of clock and the other integrated circuit MAX485 is used for transferring signal of data. The signal of clock is the output signal of the microcontroller. Integrated circuit MAX485, which is used for transferring the signal of clock, is configured as a transmitter. It is done by its second and third pin. These pins define the direction of transfer and they are connected to the high logic level and that is the voltage of 5 V. The signal of data is the input signal of the microcontroller. The integrated circuit MAX485, which transfers the signal of data, is configured as a receiver. Its second and third pin is connected to the low logic level and that is the voltage of 0 V. The microcontroller generates the signal of clock and reads serial data from the encoder. The encoder sends serial data bit by bit. The microcontroller reads input bit on the falling edge of a clock [5]. Data reading from encoder is done periodically with the period of 10 ms. The microcontroller generates signal of clock on every 10 ms. The signal of clock exists during the reading data from encoder. The duration time of pulse and pause of the clock is equal and has the value of 10 µs. It means that the period of clock is equal to 20 µs. Data length from encoder is 14 bits. A few clocks are necessary for the initialization of reading from encoder, 14 clocks for reading 14 bits from encoder and one clock after the reading is ended. The maximum 18 clocks are needed for reading position from encoder. It means that the maximum time needed for reading data from encoder is equal to 360 µs, if period of clock is 20 µs. Every 10 ms the microcontroller generates pulses for the reading data from encoder. The signal from encoder is at the high logic level in the beginning. The encoder sets the data on the low logic level when the encoder receives a clock from the microcontroller. In that way the encoder sends confirmation to the microcontroller that it receives pulses of a clock. This low logic level of data presents the message of acknowledgement. Then the encoder raises the signal of data on the high logic level. In that way the encoder reports to the microcontroller that it is ready for sending data. The microcontroller reads logic one on its input on the falling edge of a clock. In that way the microcontroller understands that the encoder is ready for sending serial data of 14 bits. This higher logic level from the encoder presents starting message. The microcontroller reads bit by bit serial data of 14 bits and packs them in the digital word. The microcontroller raises the signal of clock on the high logic level after reading of 14-bits is ended. The data from the encoder have the length of 14-bits and they are presented in the form of complement of two. This data must be shifted 2 bits in left in order to enable the operation subtraction. In that way the data of 16-bits in the form of complement of two is obtained. The next operation is the subtraction between measured value of position and zero position value. The difference between these two numbers is a number which is also in form of the complement of two. It can be positive or negative number. The values are positive in the first half of a circle and they can be in range from 0 to 32768. They respond to the value of an angle from 0 to 180°. The values are negative in the second half of circle and they can be in range from 0 to -32768. They respond to the value of angle from 0 to -180°. The value of zero position is taken during setting parameters. This value should be shifted for 2 bits in left and after that put in the EEPROM memory of the microcontroller. Every shifting digital number for 2 bits in left presents multiplying with 4. After digital data transferring from a device to a computer, it is necessary to have that number divided by 4. In that way the value of an angle from 0 to 180° responds to a number from 0 to 8192. The value of angle from 0 to -180° responds to a number from 0 to -8192. The time diagram of the signals from the encoder during data reading is shown in the figure 4. The upper signal presents the clock and the lower signal presents the data.

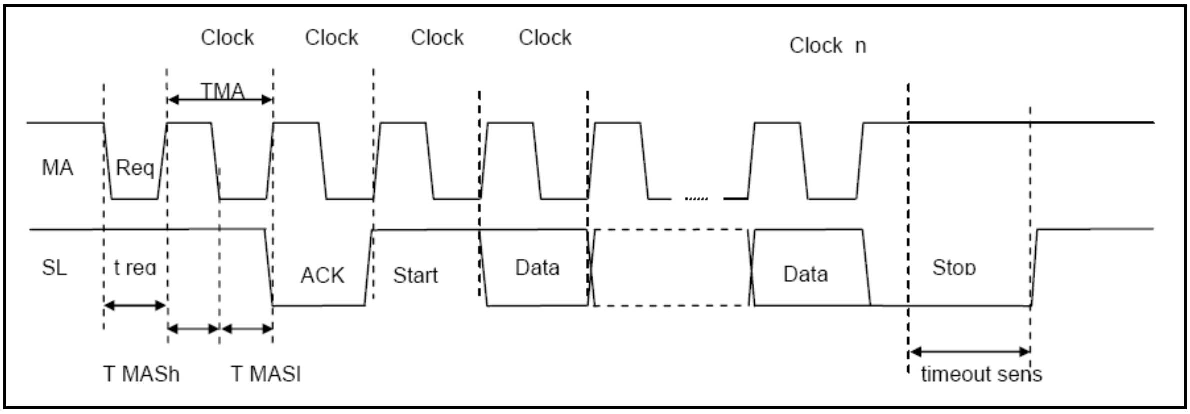


**Figure 4.** The time diagram of the signals from the encoder

## 2.3. Serial interface RS232 through computer

Communication between a device and a computer is realized by the serial RS232 interface at the baud rate of 115200 bits in seconds. This baud rate is the maximum baud rate because it is the maximum baud rate of the integrated circuit MAX232. The integrated circuit MAX232 is used for the translation of voltage levels from TTL level to RS232 level. The converter RS232/USB is used only in the case when computer does not have the RS232 interface. For serial communication it is very useful for a device to have 2 LED diodes (figure 2). The red LED diode is used for showing the data sending from a device to a computer. The red LED diode blinks during data sending. The green LED diode is used for showing data receiving from a computer to a device. The green LED blinks during data receiving. These LED diodes are very useful because in the case of some malfunction it is very easy to determine the existence of data transfer. These LED diodes are connected to the cathode with the pins of microcontroller dedicated to receiving and sending data. These LED diodes are connected to the anode with pull up resistors. The electronics for the realization of RS232 interface are presented in the figure 5.

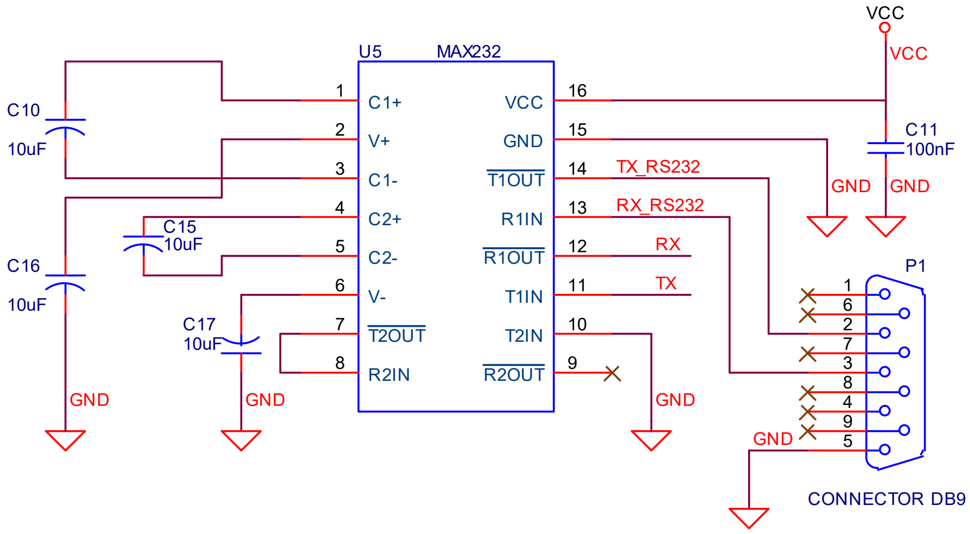


**Figure 5.** Serial interface RS232

Serial communication on the baud rate of 115200 is realized by writing value in the U1BRG register which is obtained by the next equation:

$$U1BRG = \frac{F_{CY}}{16 \cdot B} - 1 \qquad (7)$$

where is:

$F_{CY}$ – frequency of microcontroller

$B$ – baud rate of RS232 interface

If the frequency of the microcontroller $F_{CY}$ = 20 MHz and if the baud rate B = 115200 bits in seconds then the value of the U1BRG = 10.

## 2.4. Power supply

The microcontroller and the integrated circuits which are used in this device demand the voltage of 5 V for their work. The power supply of the device can be realized in two ways. The first way is that the device is power supplied by a USB port. The development board which is used in this device has the USB port. This USB port is dedicated to programming microcontroller, but it can also be used for power supply of the device. Nominal voltage of power supply of USB port is 5 V. Voltage from the USB port, according to the standard, must be in range from 4.4 V to 5.25 V [6]. The current consumption of this device from the voltage source of 5 V, which is USB port, is 270 mA. The second way of power suppling is using extern voltage source of 9V and linear regulator LM7805 which is resided on the board. Linear voltage regulator LM7805 is used for getting voltage of 5 V. The figure 6 shows the electronics for power supply of a device.

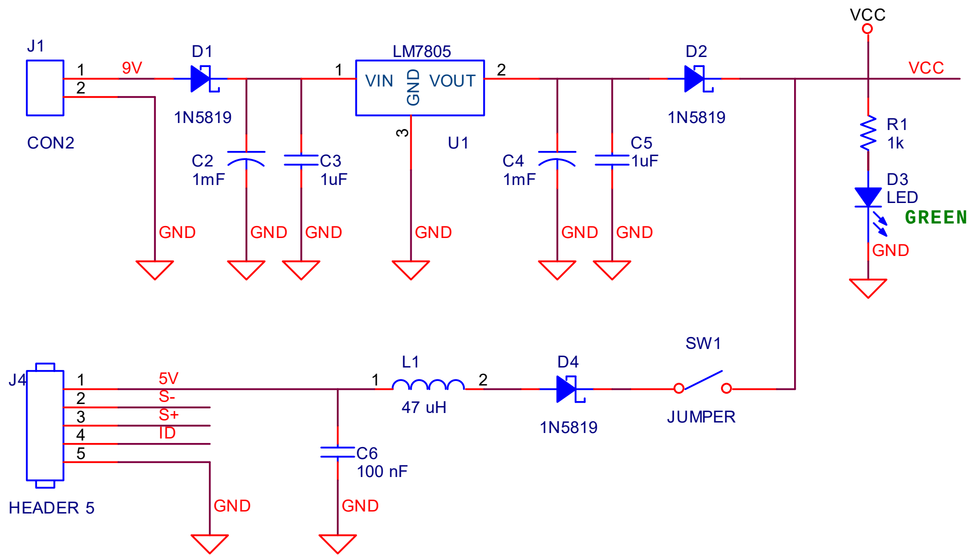


**Figure 6.** Power supply of the device

Microcontroller dsPIC30F6014A has integrated Brownout detector which can be turned on by configuration bits. The brownout detector is an electronic circuit which measures the voltage of power supply of the microcontroller. If voltage of power supply of the microcontroller drops below defined value, the brownout detector resets the microcontroller and keeps it in the reset state until the power supply voltage of the microcontroller rises above the defined value. The minimum voltage of the power supply is asserted by the configuration bits. It is very useful to turn on Brown detector because in that way innacuarate performance of the microcontroller is avoided. In this case, the minimum voltage of power supply is asserted to 4.2 V.

## 3. SOFTWARE APPLICATION

Software application for communication between a device and a computer is written in C# program language. The execute program is run by clicking on the file "Encoder.exe". After that the main window is opened. The window for setting parameters of serial port is opened by pressing on the button "Open connection". The window for setting parameters of the serial port is shown in figure 7.

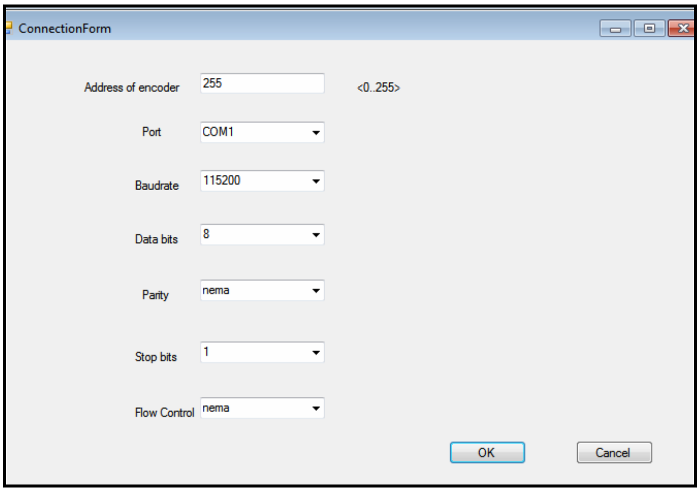


**Figure 7.** Setting parameters of the serial port

In this window a user only chooses the name of a serial port "COM". It is predicted that every device has its own address which can be in range from 0 to 254. The address which is equal to 255 is the common address for all devices and it is used if a user forgets the address of a device. The main window for communication with the device is shown after oppening of a serial port. There are 5 commands which can be used for communication. At the beggining, it is necessary to determine if the communication between a computer and a device exists. It can be checked by command "LINK RESET". The device gets this command and sends a message of acknowledgement by which the device informs the user that the communication between the device and the computer exists. The computer waits for the device to send the acknowledgement message. If the device does not send the acknowledge message in 1 second, then the computer will write the message "timeout" on the monitor. This case can happen only if the wire between the device and the encoder is cut off. By pressing the button "SET PARAMETERS" a window is opened for setting the address of a device and zero position of the encoder. The window for parameters setting is shown in the Figure 8.

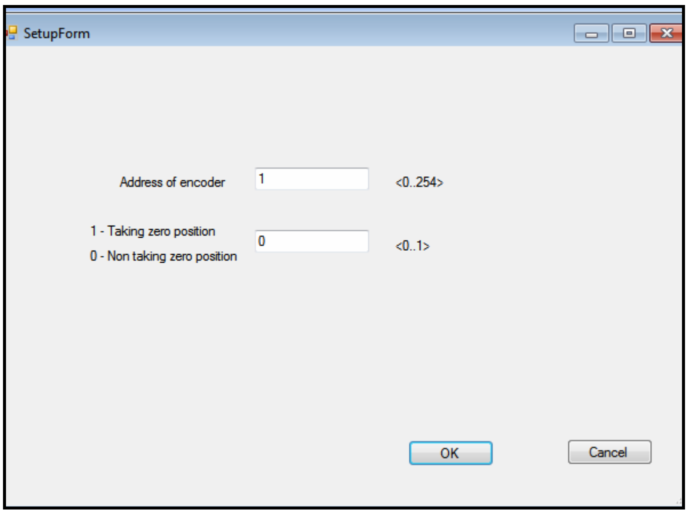


**Figure 8.** The window for setting the parameters of a device

The choice for setting zero position is given. The address of a device will be set in every case, but a user can decide if they want to set zero position. The address of a device and the zero position of the encoder are written in the EEPROM memory of the microcontroller which means that they remain even after the power is turned off. The zero position of the encoder is set only at the begining of measuring. Firstly it is necessary to set the encoder in the zero position. Then it is necessary to choose an option for setting zero position of the encoder in the application program. The device reads value from the encoder and puts it in the EEPROM memory of the microcontroller during setting zero position of the encoder. The device resets itself after the parameters are written in EEPROM memory. Then the program goes in the infinite "while" loop from which can not go out until the watchdog timer does not reset the device. The timeout of watchdog timer is set to 128 ms. The instruction for clearing of watchdog timer is written in the main program on the beginning of the main "while" loop. The watchdog timer works on the frequency of the RC oscillator which is integrated into the microcontroller. During the program instalisation, which occurs after reset, the value of parameters is read from EEPROM memory and they are written in variables which occur in RAM memory. The main program works with these variables from RAM memory. At every time a user can read the asserted address of the device by clicking on the button "READ PARAMETERS". A user can reset the device by clicking on the button "RTU RESET". A user can read the position from encoder by pressing button "READ DATA". By pressing on the button "HELP" a manual for work with the application is opened. A user should press the button "Close connection" after the work is finished and then the serial port is closed. The main window of application is shown in the figure 9.

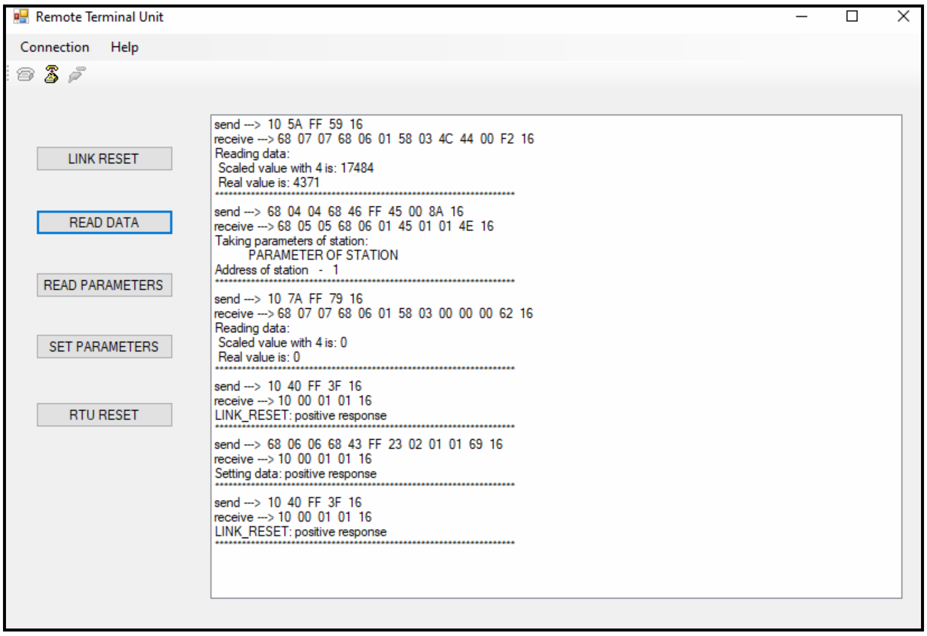


**Figure 9.** The main window of application

## 4. CONCLUSION

In the most cases the resolution of 14 bits of angle is quite satisfactory. It means that the minimum change of angle, which can be detected, is equal to 0.022 degrees or 1.32 minutes. The reason for this is that the angle of 360 degrees, which is a full circle, is presented with the data which is equal to 16384. The number 16384 is got as $2^{14}$. This device offers further development in the way of serial communication. The serial communication RS232 becomes absolute, because a very small number of desktop computers have the interface RS232. Today it is very hard to find a laptop computer which has the interface RS232. The converter from RS232 to USB must be used if a computer does not have the interface RS232. Further improvement can be done in the way of developing a device with the USB interface. In that case the converter RS232/USB should not be used which brings low cost of device.

## References

[1] MICROCHIP: *dsPIC30F6014A Datasheet*, Microchip Technology, 2011.

[2] MICROCHIP: *dsPIC30F Family Reference Manual*, Microchip Technology, 2006.

[3] MICROCHIP: *dsPIC30F Digital Signal Controllers*, Microchip Technology, 2005.

[4] MAXIM INTEGRATED: *MAX485 Low Power RS-485/RS-422 Transceivers,* datasheet, Maxim Integrated Products, 2014.

[5] HENGSTLER: *ACURO – SSI/BiSS*, Technical Datasheet Interface.

[6] AXELSON,J.: *USB Complete*, page 436, 2005.