Wrocław University of Science and Technology

Electronics Department — Faculty: Embedded Robotics

Intermediate Project Final Report

Position monitoring system for climbing robot

Author
Karol Duda

Supervisor
Dr. Witold Paluszyński

Abstract

Main project goal: to make robot able to measure it's position.
Main assumptions that system has to fulfill: it has to be very light (in order not to interfere with normal robot operation), it has to be accurate ($< 0.1\%$ of error) and cheap (affordable for student).
Main results obtained: a complete system for one leg position measurement has been done. Although obtained error values in system tests was acceptable, it occurred that biggest part of the error comes from sensor board imperfection. Hence, conclusion is that obtained results can be improved and it has to be done before making system for whole robot. Additionally ability to send measured data by bluetooth protocol has been implemented. This feature makes us able to connect robot to some external controller in order to control it's movement.

January 25, 2018

This work is licensed under a Creative Commons "Attribution-NonCommercial-ShareAlike 3.0 Unported" license.

1 Introduction

Project refers to the robot climbing flat vertical surfaces which has been made for Engineer's Thesis. Although the climber achieved its goal and made first steps on the window, it requires many improvements. Example of required improvements is system mentioned in project title. Main goal of this project is to create and test such a system. The robot itself is shown on the figure below:

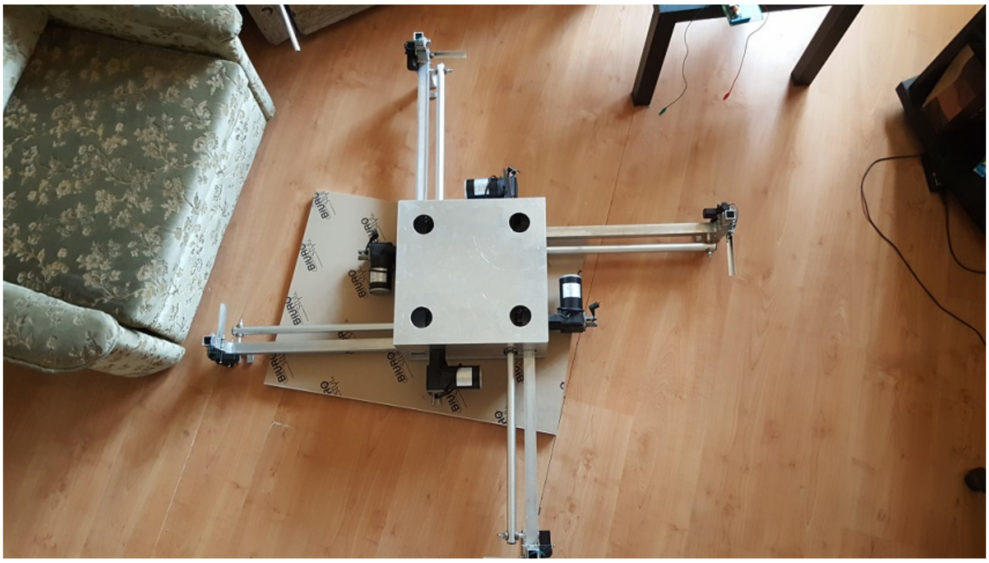

Figure 1: Vertical surfaces climbing robot

By position monitoring we understand here measuring only two axis data on the working plane. Such idea is shown on the figure below:

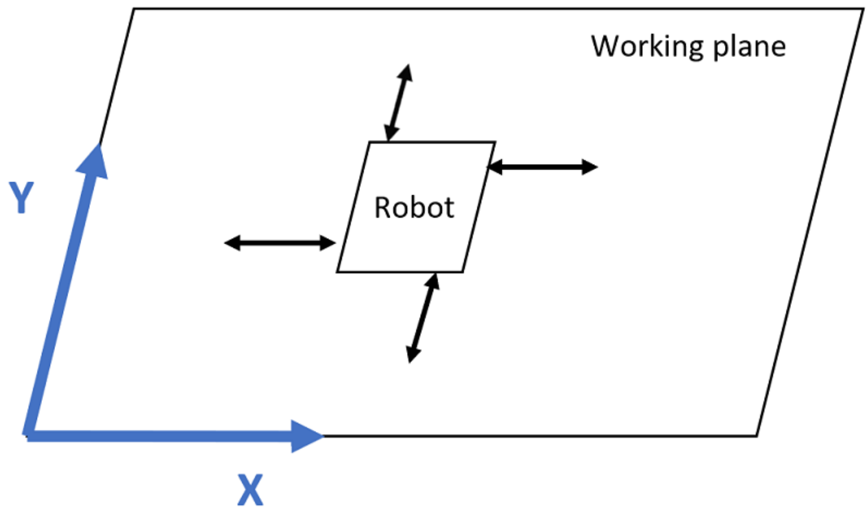

Figure 2: Measurement idea schematic

As we can see the point is to get the X and Y values of the robot position in relation to some reference (i.e. starting point which X and Y values are known). Mechanical structure of the robot allows us to measure this data by monitoring four vertical legs movement. Resulting position would be calculated by incrementally adding next measures. This system should be enough accurate to obtain reliable information after few robot steps in order to be able to give robot some command in the future like:"move 25,6 cm left".

2 Realisation

2.1 Goals to achieve

List of most important tasks is following:

- Specify all of the system features and functions
- Check out the possible solutions and choose the most suitable.
- Design the PCB fitted to requirements
- Make the prototype
- Setup all elements on the prototype
- Setup prototype in the robot
- Implement the communication with microcontroller
- Tests (with microcontroller in debug mode)
- Compare data quality from different sensor outputs (SSI,PWM,Quadrature signal)

2.2 Chosen measurement method

After considering many measurement methods like optical measurement,mechanical measurement or magnetic measurement, it occured that last one mentioned is most suitable for our requirements. Magnetic linear movement measurement is relativly cheap (affordable for a student) very reliable (in comparison to optical encoders) and accurate. After research in available solutions I have chosen the AS5311 sensor from AMS company. It uses the Hall effect to obtain contactless measure of linear motion. It's resolution is up to 488nm (12bit for 2mm) and accuracy about single μm ($\pm 10\mu$m max) what makes it much more accurate than we actually need. Sensor is designed to coperate with magnetic strip of pole length 2mm in order to achieve its best accuracy. Another advantage of this IC is that it has 3 different outputs, which are namely:

- Digital output (SSI non-differential interface)
- PWM output (Duty cycle proportional to position of the magnet)
- Incremental outputs with index pin

First two outputs have 12 bit resolution for polepair and Incremental outputs have 10 bit resolution (1024 steps for polepair) what reduces its accuracy. Principle of sensor operation is shown on the following figure:

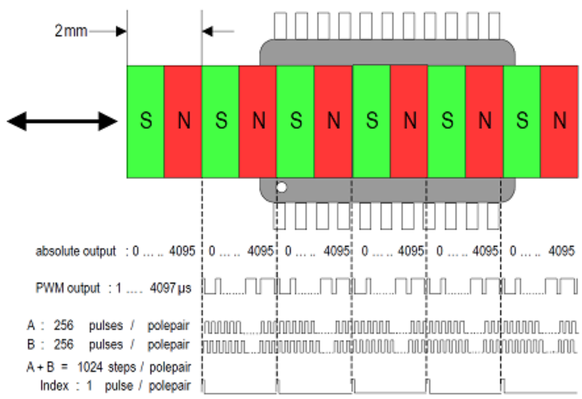

Figure 3: Sensor principle of operation

Another advantages of this chip are: small price, small case(TSSOP 20)and low power consumption (70mW max). Disadvantages are: very precised working set (0,2mm vertical air gap between sensor and magnet required, maximum 0,5mm shift between IC centerline and center of the magnet is recommended), SSI digital interface which is very rare implemented in microcontrollers.

2.3 Hardware

2.3.1 Electronic hardware

In order to use AS5311 sensor as a measurement device in our robot, dedicated PCB is required. Such board would be IC extension providing: an input of supply voltage, outputs of measurement data which have to be send to controller of the robot, assembly possibility and necessary external elements (i.e. capacitors). It have to be compatible with robot controller already existing. Controller have four cable sockets dedicated to leg movement measurement. These sockets and their signals are included in figure 4.

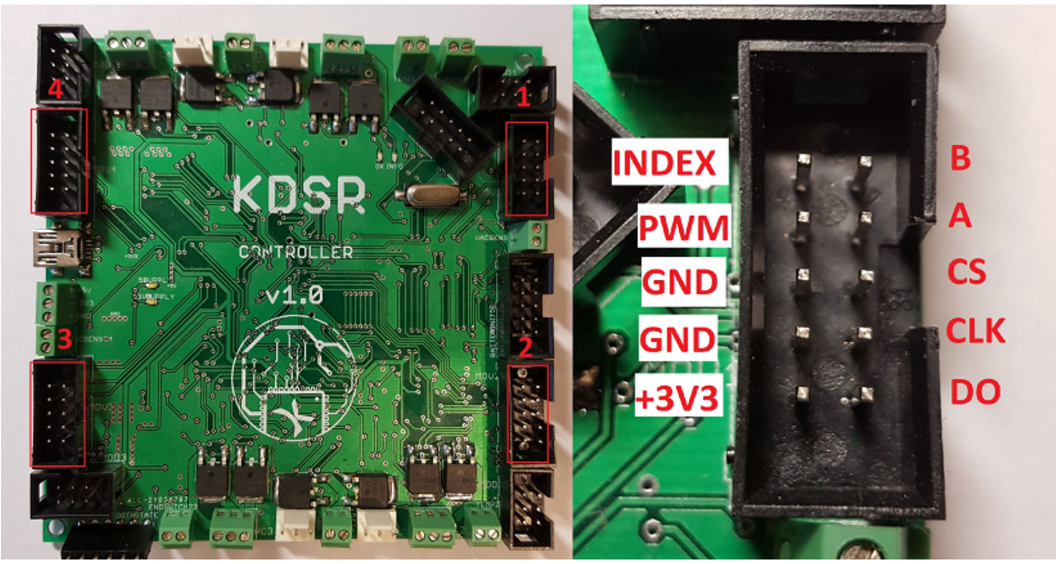

Figure 4: Controller board with dedicated sockets

Where:
– Numbers above sockets indicate which leg are they intended for (legs are numbered clockwise)
– INDEX,A,B are inputs for quadrature signals from AS5311
– PWM is input for Pulse Width Modulation signal from sensor
– CLK is serial communication clock signal (output)
– CS chip select (output)
– DO data output signal from sensor (input)
– +3V3 and GND are power supply for sensor board (outputs)

Sensor board had to be small in order to fulfill the hardware requirements which are mentioned further in section 2.3.2. Design of the board and board with soldered elements is shown on the following figure:

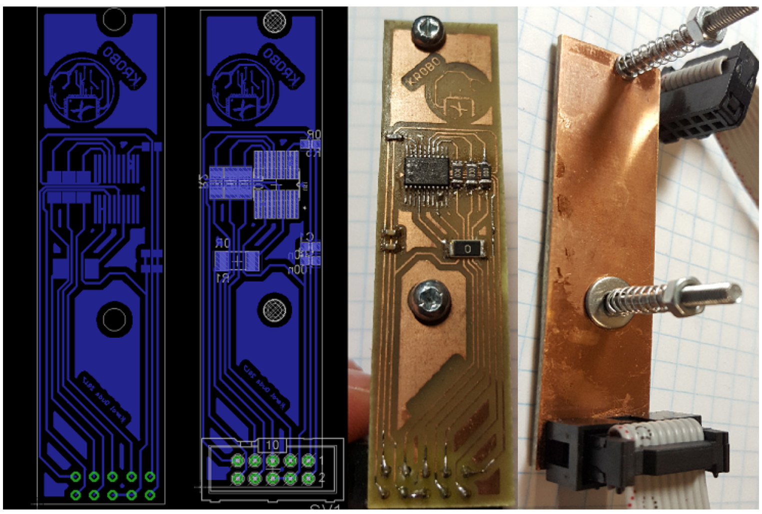

Figure 5: Board project and ready board with soldered elements

2.3.2 Other hardware

Positions of the holes for screws and chip on sensor board could not be random since there is only 0,5mm admissible shift between center of the magnet and center of the sensor. Magnet is placed on the middle of leg on its outer side in order not to interfere with leg movement rollers. Magnet mounted on robots leg is shown on figure 6.

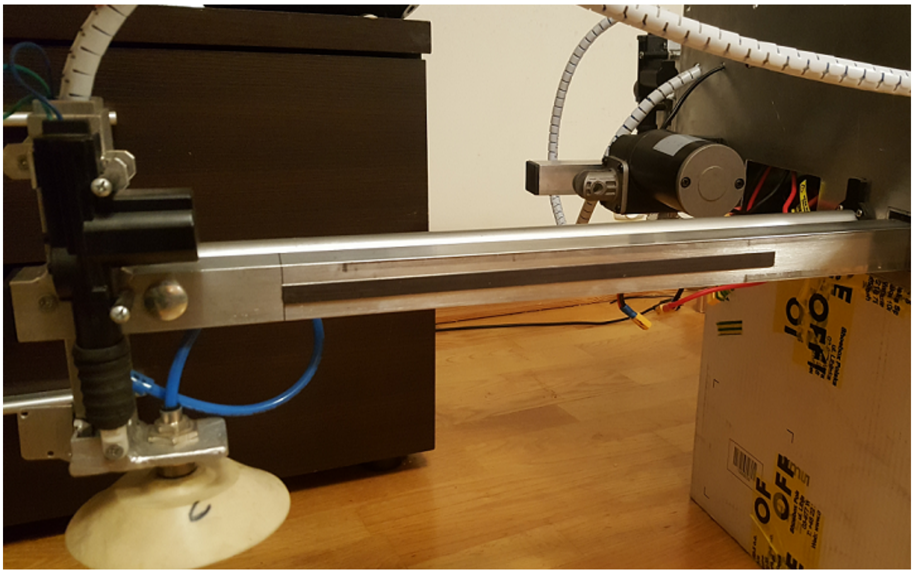

Figure 6: Magnet mounted on robots leg

Hence, next task was to mount sensor board in such way that IC center line would be exactly matching the magnet center and vertical gap between them wouldn't exceed 0.2mm. In order to achieve that, dedicated board holder has been made. Mounted holder is shown on figure 7.

Figure 7: Board holder mounted in robot

On the figure 7 we can see also marked starting point of the magnet strip. It is worth mentioning that leg is captured in it's zero position on this picture, so magnet protrudes about 1.5 cm from the robot what is actuators "zero value". That's exactly one of the facts which sensor board is adjusted to. With currently known hardware we can be sure that horizontal and vertical position of magnet with reference to chip will be correct, but another important feature is space between them. Sensor datasheet contains information that it should be 0.2cm and should not exceed 0.3cm. My solution to this requirement is to use springs which will constantly push sensor board to the magnet. In order not to make distance between chip and magnet zero (contact) and to make board short-circuit proof it is all painted with clear varnish. Whole mounted measuring system is shown on the figure 8.

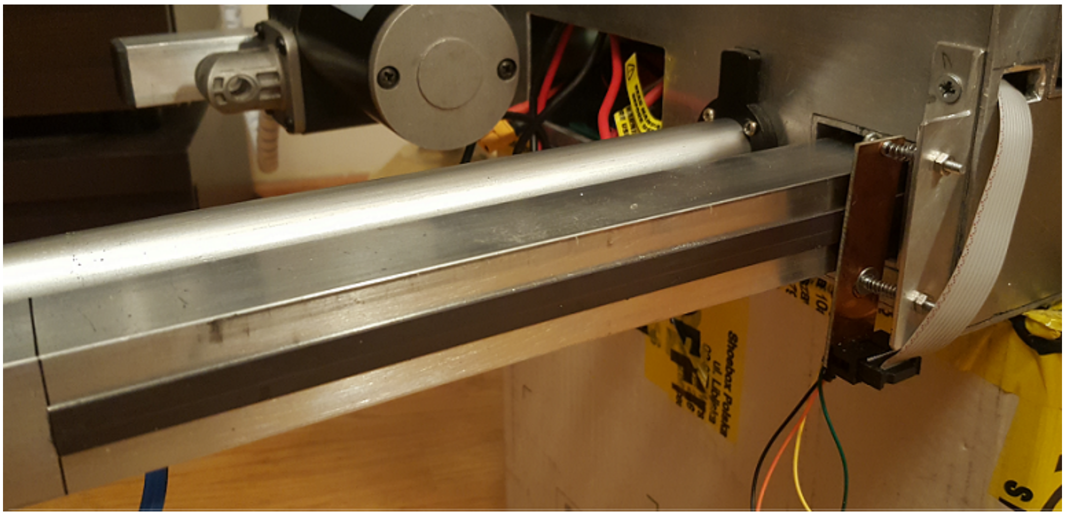

Figure 8: Whole measurement system ready to work

2.4 Software

2.4.1 Communication

The most important way to get data from the sensor is in this case is digital SSI data protocol because of its accuracy and reliability. SSI is Synchronous Serial Interface which is generally differential but in our sensor it is some non-standard version which is non-differential. Unpopularity of such protocol makes it almost impossible to find the microcontroller which has it implemented. That's why it was necessary to implement this protocol on normal GPIO (input/output pins).
There are two kinds of data frame in this protocol, one with position data and one with magnetic field strength data. Communication operation when position data is sent is shown on figure 9.

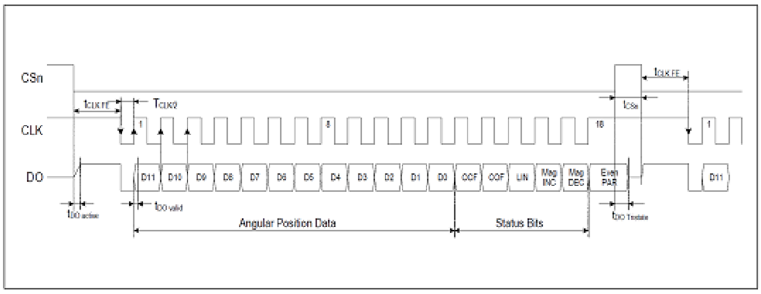

Figure 9: Operation of communication with position data frame

Communication operation when magnetic field strength data is sent is shown on figure 10.

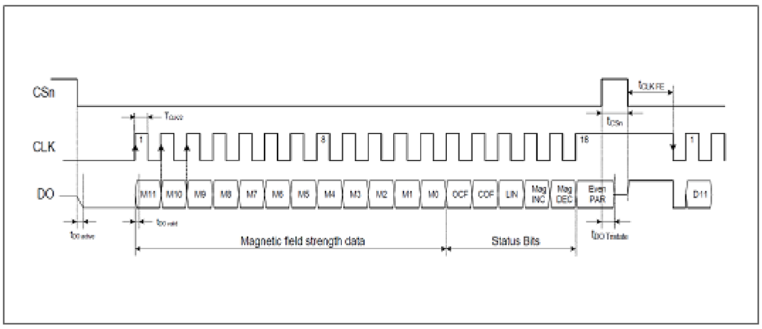

Figure 10: Operation of communication with magnetic field strength data frame

Only difference between these two types (despite different data to be send) is clock polarization. In this project it is necessary to use both of them. Position data is needed to receive measures values, magnetic field strength is needed to make sure that they are reliable. Implemented function which is responsible for receiving the position value is shown on figure 11.

```
void measure_position(){
    volatile int i;                          // counter variable initialization
    uint8_t checksum_even=0;                 // variable containing data needed for checksum initialization
    datauint=0;                              // reset of the variable containing position data
    cs1_PutVal(0);                           // start of communication (Chip Select-active low)
    for(i=0;i<10;i++){}                      // small delay (required in datasheet )
    communication_started=TRUE;              // set the flag which makes the clock running (toggle the clock pin state in interrupt)
    for(i=17;i>=0;i--){                      // main communication loop
        while(clock_high!=FALSE){}           // wait until the clock state is low
        while(clock_high==FALSE){}           // wait until the clock state is high
        if(clock_high){                      // if the state of the clock is high
            if((do1_GetVal())==1){           // if value on DO pin is high
                datauint|=(1<<i);            // save it in the variable
                if(i!=0)
                    checksum_even++;         // and increment checksum counter
            }
        }
    }
    communication_started=FALSE;             // stop the clock generation
    control_bits=(datauint&0x3f);            // get the control bits from whole communication result
    if(((control_bits & 0x20)==0x20) && ((control_bits & 0x10)!=0x10) && ((control_bits & 0x8)!=0x8) && ((checksum_even%2)==(control_bits & 1) )){
        position_value=(0xFFF&(datauint>>6));    // if data is correct -> save it to main position variable
        new_measure=TRUE;                        // set the flag which indicate that new measure has been done
    }

    clk1_PutVal(1);                          // make sure clock is in high state
    clock_high=TRUE;                         // set the high state of the clock flag
    cs1_PutVal(1);                           // end the communication (Chip Select to high)
}
```

Figure 11: Function responsible for communication with the sensor (postition data type)

This function is continuously called by periodic interrupt with frequency of 2 kHz. Clock is generated by second periodic interrupt with frequency 200 kHz after the global variable *communication_started* is set. Function responsible for receiving magnetic field data has been implemented in the same way with only difference in clock polarization. Those two functions provide new data in specific global variables few thousand times per second. In the main application file after *new_measure* global flag is set, position value is added to integrated position variable which provides information about current leg position.

3 Results

After integration of all modules, first tests of the system took place. Tests has been done on the controller in debug mode, so we could know in real time what is the calculated value of leg position and measure the real value. Although these test seems most intuitive, it is very hard to compare result from measuring tape (with accuracy of approximately ±0.5mm) and our system (with accuracy of approximately ± 10μm). Such tests has been done and their results seemed that system work correctly. To obtain more reliable information about the system work, tests with one static position value were done. Static value was the "zero position" of the actuator, what helped us to measure the error of integrated position. Results of these tests are shown on figure 12.

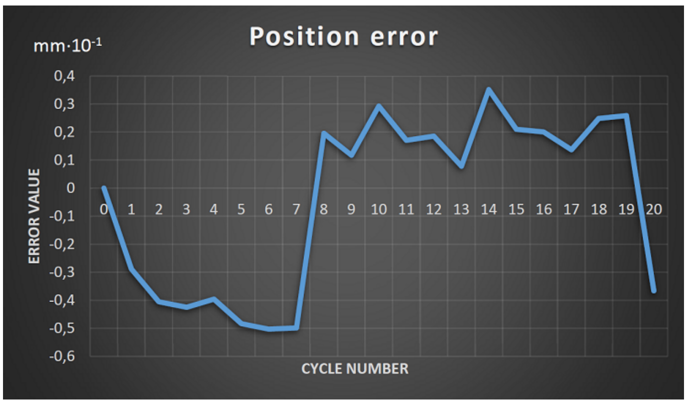

Figure 12: Integrated error chart after specified number of full actuator cycles

As we can see on figure 12 values of integrated error are very small. Greatest error value was obtained after 7th actuator cycle and it reached value of -50μm and summarized actuator motion length was 420 cm calculated from following equation:

$$leg\_motion\_length = number\_of\_full\_cycles * (2 * maximal\_actuator\_stroke) \quad (1)$$

That lead us to conclusion that percentage of error in the final result is very small. Although results are satisfiable, there occurred a problem which make the values on the plot 12 seems like having some inconstant bias. The problem is sensor board imperfection. This PCB was drilled by simple drill what makes the holes not perfect. During tests occurred that board is moving a little on the screws because of this fact. It is probably source of error non gaussian distribution (so error most significant part is created by this problem). After noticing it I have made new sensor board project (with some corrections) and ordered professionally made PCB's. Unfortunately their time of delivery is about few weeks so I wont be able to test them and made whole positioning system as a part of this project.
Additionally a function which sends the leg position data by bluetooth has been implemented. This part of code sends the data by UART protocol to connected HC05 module with 6 decimal digits of accuracy. Test of such functionality is shown on figure 13

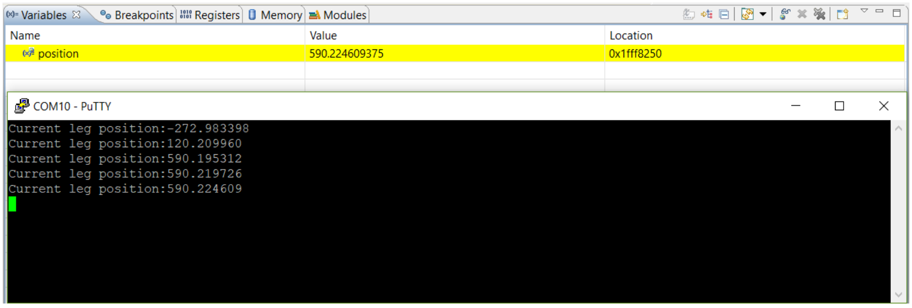

Figure 13: Tests of sending the data by Bluetooth

Upper part of figure 13 is one of the environment debug windows which makes us able to see current global variables values and on the bottom is Putty application window which shows currently received data stream. Yellow colour of the variable line indicates that value has recently changed because leg was still in move. As we can see sending the data works perfect.

4 Tools

In this project I have used following tools:

- Cadsoft Eagle 7.2.0 (trial version)
- Code Warrior 10.7 (trial version)
- J-link Programmer (edu version)
- Saleae Logic analyzer
- Own designed controller of the robot with MK40DX256VLQ10 microcontroller
- PuTTY application

5 Resources and literature

- Project using HAL effect sensors to measure the linear position
  http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6750978
- Project with similar goal with usage of expensive high speed camera
  http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8063194
- Microcontroller MK40DX256VLQ10 datasheet
  https://www.nxp.com/docs/en/reference-manual/K40P144M100SF2V2RM.pdf
- Position sensor documentation
  http://ams.com/eng/Products/Magnetic-Position-Sensors/Linear-Position/AS5311
- Magnetic strip documentation
  http://ams.com/eng/Products/Magnetic-Position-Sensors/Magnets/AS5000-MS10-300
- Saleae logic analyzer usage tips
  https://www.saleae.com/