An approach of developing low cost ARM based CNC systems by controlling CAN drives

Georgi M. Martinov[1], *Akram* Al Khoury[1,*] , and *Ahed* Issa[1]

[1]FSBEI HPE MSTU “STANKIN”, Moscow, Russia

Abstract. Nowadays, there is a big demand on using small sized CNC machine tools, which have low price tag, wide range of implementations, low manufacturing costs and can be used for educational purposes. These machines can achieve casual manufacturing routines, like milling and drilling in applications, where there is no need for high speed performances and super quality of products. In this work, we proposed a model of CNC for these machines and analysed its components and efficiency. The model consists of three main layers: CNC system (application layer), ARM based microcomputer as CAN master and controller (connecting layer) and Servo-Drive Step Motors (actuating layer).

1 Introduction

Servo-Drive Step Motors are traditionally considered as one of the important basic elements of technological systems for various purposes, which used in most industries. Due to their reliability of the drives that determines degree of production efficiency and level of general safety of industrial machinery.

Servo-Drive Step Motor is a high-performance servo drive with vector control for speed and torque. This servo drive is intended for the widest possible application in engineering and other industries.

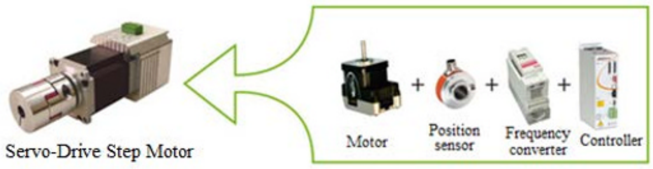

Fig. 1. Servo-Drive Step motor structure.

Structurally, the Servo-Drive Step Motor consists of the following main units:

1. Hybrid stepper motor with NEMA 23 and 34 dimensions.
2. Frequency converter based on a high-performance DSP processor.
3. Control unit (servo controller and programmable logic controller in one block).
4. Motor shaft position sensor.

[*] Corresponding author: akramalkhoury@gmail.com

The CNC system is equipped with means for setting up technological complexes, connecting to the workshop networks. Open Modular Scalable Management Architecture allows machine tool makers and end users to expand its functionality by simply integrating new software and hardware solutions.

These specifications lead us to find a way to implement these servo drives in CNC Systems, as they support CAN protocol. Here, we are looking for technical solution that can deliver correct commands from CNC controller to these motors.

Hence the main CNC controller must have these requirements to ensure the connection mechanism of third-party CAN equipment:

1. The ability to connect industrial devices of various kinds (input-output devices, controllers and drives).
2. Connection between CNC controller and CAN master accomplished using Ethernet TCP/IP standard.
3. Cycle time of data exchange in the network (TCP/IP cycle) doesn’t depend on the cycle time of any terminal device (CAN cycle).

2 Goals of the work related to CNC

As was mentioned before it is necessary to ensure connection mechanism between the high-level (CNC controller) and low-level (drive). This mechanism should be reliable, inexpensive, easy implemented and flexible.

Currently, the solutions that are available on the market are expensive and have excessive functions and most of them are closed. This leads to the complexity and inability to use existing solutions for a cheap segment of industrial production.

Nowadays microcomputers and microcontrollers play an important role in open source programming. Also, they have a wide range of implementations due to their ability to be adapted in almost any part of applications.

We are interested in the family of microcomputers Raspberry PI3, as it supports a wide range of communication protocols such as built in Wi-Fi, Ethernet, Bluetooth and SPI adapters. Raspberry PI3 also has 40 GPIO pins, that allow us to connect other peripherals, which add supplemental functionality.

The idea is to use the Raspberry PI3 as intermediary device i.e. it is a way that ensures the connection between the CNC controller and Servo-Drive Step Motors. Here, the discussion will be about the hardware and the software solution. Raspberry PI3 can run an independent operation system, which make it a powerful tool to handle program threads and other type of synchronization procedures.

Realization of physical layer that ensures data exchange needs a hardware solution, so the schematic chip MCP2515 was used. MCP2515 is a standalone Controller Area Network (CAN) controller, that implements CAN specification v.2.0B. The MCP2515 interfaces with the Raspberry PI3 via an industry standard Serial Peripheral Interface (SPI). Transceiver XCVR will be the schematic chip TJA1050, which is an interface between the Controller Area Network (CAN) protocol controller and the physical bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller.

3 Used protocols to control Servo-Drive step motor using CAN

The program solution will include necessary structures, which generate corresponding and packets. Those packets request needed control information from drive and rewrite it using another format to answer the CNC controller and vice versa, i.e. getting packs from CNC to set up the needed parameter (speed, torque, PID constants, etc.) and forming the appropriate packet to send it to CAN device.

Usually parametrization device will be first step. In this step, Technical CAN protocol is implemented in the stage of parametrization. Then starts cyclic data exchange procedure, that allows to control the step drives, which are connected to CNC controller.

There are two types of protocols that ensure the connection with the motors: CAN protocol and Technical CAN. CAN protocol is used as real time protocol. Technical CAN protocol is implemented in the stage of parametrization.

3.1 CAN protocol

The current implementation uses a standard message format consisting of 11-bit identifier (ID0-ID10) and data, the length of which varies from 0 to 8 bytes. This protocol will be used in cyclic data exchange stage, that ensure continuously update working data parameters for each drive and input-output device.

Table 1. Format message using CAN protocol.

ID10 – ID6	ID5 – ID3	ID2-ID0	8*8 bits
Command	Source address	Sink address	Data

Within the same network, each device must have unique address in the range from 0 to 6. Address 7 is used as a broadcast address. Any parcel transmitted with a value of source address of 7 will be accepted by all drives (broadcasting).

Bitrate of data exchange will be up to 1 Mb/s corresponding to ISO 11898-2.

3.2 Technical CAN protocol

The technological protocol is intended for setting up the drive and analyzing its operation. With this protocol, the user can access both write and read to any parameters.

The process protocol is transmitted to the data field D0-D6 of the drive control report message via the CAN bus.

This protocol will be used to read and setup all kinds of parameters like (maximum speed, torque, PID constants, etc.)

4 Architectural model of used solution

Proposed model consists of CAN master realized on the base of C program, which is implemented on Raspberry PI3. This program ensures the connection between CNC system and motors. Packets which came from CNC Control system through Ethernet and corresponding messages, will be formed in Raspberry PI3 to send them to motors via CAN bus.

Next structure was used to form the desired CAN message, that will be sent to the servo motor. Where id is the CAN 11-bit identifier and data is 8*8-bit CAN data.

```
struct CAN_MSG {
unsigned int id;
unsigned char data[8];
};
```

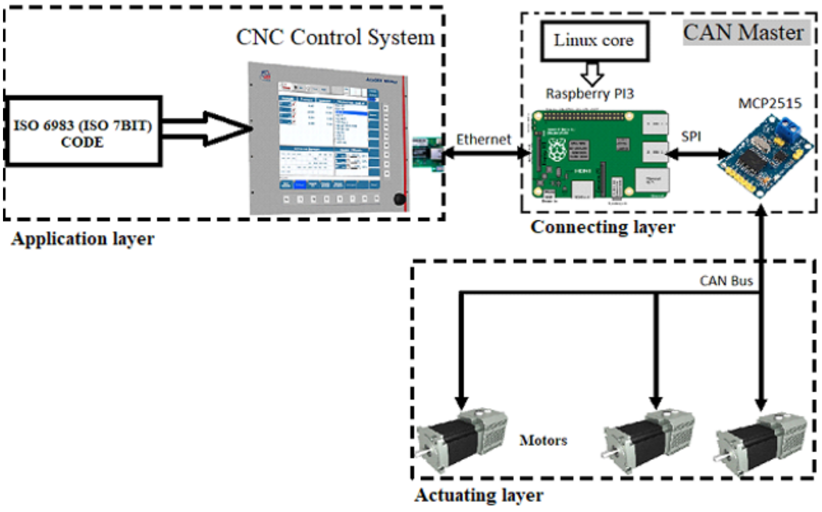

Fig. 2. System diagram.

Schema connection MCP2515 with Raspberry PI3:

```
GPIO CAN
HEADR SIGNAL MODULE
PIN NAME --- SIGNAL
#01 3,3V --- VCC
#02 5V --- Extra wire to Vcc TJA1050
#06 GND --- GND
#09 GND --- GND to CAN BUS
#21 MISO --- SO
#23 SCLK --- SCK/CLK
#24 SPI0.CE0 --- CS
#32 GPIO12 --- INT
```

Raspberry PI3 plays a role of CAN controller, its mission is forming the required packets, sending them to the motors and getting the appropriately answering packet from them.

The corresponding functions that realize these procedures are:

```
   void speed_CAN_protocole (int des,int REQUIRED_SPEED); //Set speed using
CAN protocol.
   void speed_technical_CAN (int des,int REQUIRED_SPEED); //Set speed using
technical CAN.
   void write_rasp (struct CAN_MSG can_msg); //Send the frame from raspberry
PI3 to motors.
   void read_status (struct CAN_MSG can_msg); //Read status from the motor
(request answer).
```

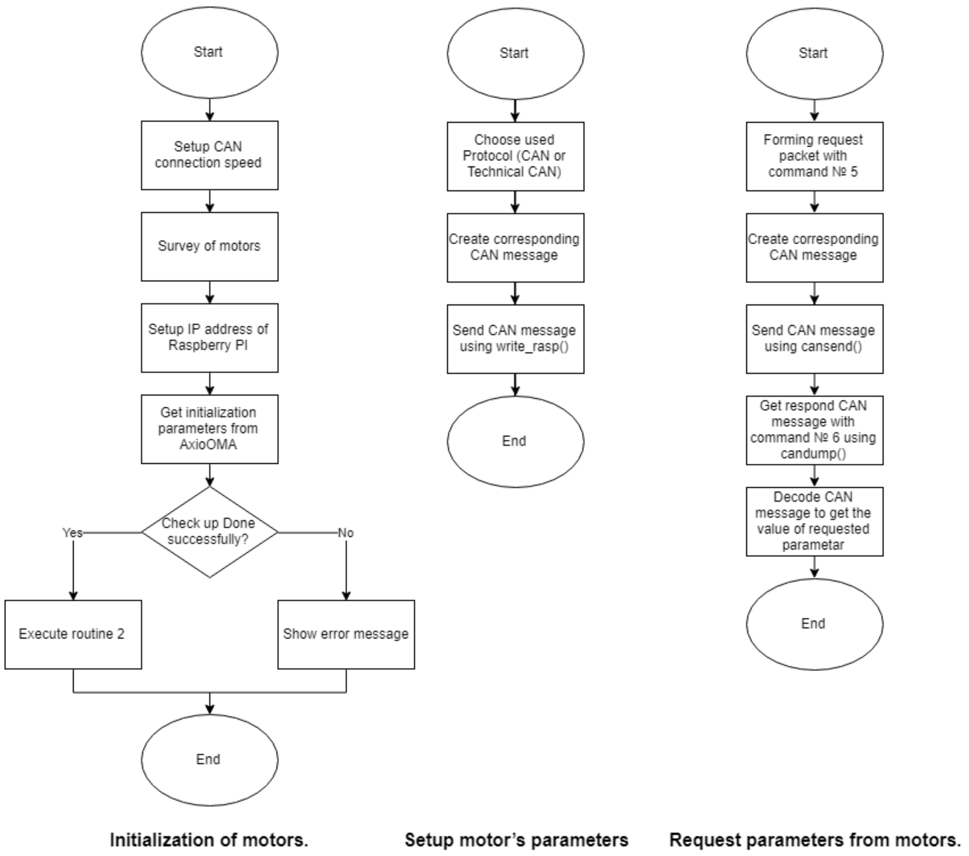
Start
Setup CAN connection speed
Survey of motors
Setup IP address of Raspberry PI
Get initialization parameters from AxioOMA
Check up Done successfully?
Yes
No
Execute routine 2
Show error message
End
Initialization of motors.
Start
Choose used Protocol (CAN or Technical CAN)
Create corresponding CAN message
Send CAN message using write_rasp()
End
Setup motor's parameters
Start
Forming request packet with command № 5
Create corresponding CAN message
Send CAN message using cansend()
Get respond CAN message with command № 6 using candump()
Decode CAN message to get the value of requested parametar
End
Request parameters from motors.

Fig. 3. Case diagram of machine state.

5 Conclusions

The connection established between Raspberry PI3 and Servo-Drive Step Motors using the CAN adapter MCP2515, which has 8 MHz oscillator.

Tests had been done by sending 100 requests (from Raspberry PI3 to the drives) to get information about the current speed of them and receiving 100 responds (from the drives to Raspberry PI3) containing data about current speed.

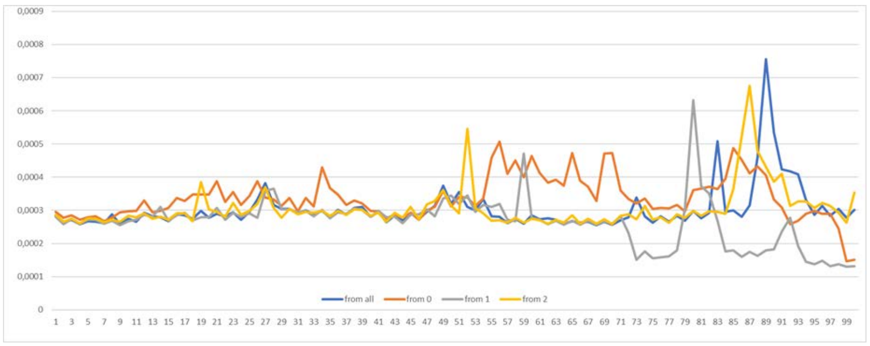

Fig. 4. Responding times for 100 request-respond cycles.

As shown, the minimum time of request-respond cycle is 0,00013 seconds, the maximum time is 0,000756 seconds and the average time is 0,00030266 seconds.

Time character of these cycles depends on many factors including Raspberry PI3 itself, drives responding and MCP2515 CAN driver.

This research was supported by the Ministry of Education and Science of the Russian Federation as a public program in the sphere of scientific activity (N 2.1237.2017/4.6).

References

1. Martinov, G.M., Martinova, L.I. *Trends in the numerical control of machine-tool systems*. Russian Eng. Res. **30**(10), 1041–1045 (2010)
2. Grigoriev, S.N., Martinov, G.M. *Control and diagnosis of CNC machine tool digital drives.Kontrol'*. Diagnostika **12**, 54–60 (2012)
3. Martinov, G.M., Lyubimov, A.B., Bondarenko, A.I., Sorokoumov, A.E., Kovalev, I.A. *An approach to building a multiprotocol CNC system*. Autom. Remote Control **72**(10), 345–351 (2015)
4. Martinov, G.M., Ljubimov, A.B., Grigoriev, A.S., Martinova, L.I. *Multifunction numerical control solution for hybrid mechanic and laser machine tool*. Procedia CIRP **1**,260–264 (2012)
5. Bushuev, V., Evstafieva, S. and Molodtsov, V. : *Control loops of a supply servo drive*. Russian Engineering Research, **36**(9), pp.774-780. (2016)
6. Martinov, G.M., Kozak, N.V. *Numerical control of large precision machining centers by the AxiOMA contol system*. Russ. Eng. Res. **35**(7), 534–538 (2015)
7. Grigoriev, S.N., Martinov, G.M. *The control platform for decomposition and synthesis of specialized CNC systems*. Procedia CIRP **41**, 858–863 (2016)
8. J. Schill, *An overview of the CAN protocol*, Embedded System Programming **10**, no. 9, pp. 46-62 (1997)
9. K. Etschberger, *Controller Area Network*, Germany:IXXAT Press, (2001)
10. CHEN Xiong-wei, *Application of Simulated SPI Based on CAN Controller MCP2515-I/SO* ; Computer Knowledge and Technology **26** (2009)
11. Pan Tong, Ye Xiaorong, Zhang Na, Fan Jianfeng, *Design and Application of CAN Bus Repeater Based on STM32 Microcontrollers & Embedded Systems* **01** (2011)
12. Chen Zujue, Zhou Ming, *The drivers design of CAN controller based on the embedded Linux*. Computer design and engineering (2006)
13. Martinova, L.I., Sokolov, S.S., Nikishechkin, P.A, *Tools for monitoring and parameter visualization in computer control systems of industrial robots*. In: Tan, Y., Shi, Y., Buarque,
F., Gelbukh, A., Das, S., Engelbrecht, A. (eds.) ICSI 2015. LNCS, **9141**, pp. 200–207. Springer, Cham (2015)
14. *CAN Specification*, 1991.
15. Servo motor drive User manual: http://www.servotechnica.ru/catalog/type/brand/index.pl?id=18