Technical support in seed production of agricultural plants

M.A. Litvinov[1,*], *A.A.* Adamyan[1], and *Hoang Ngnia* Dat[2]

[1]Federal Scientific Agroengineering Center VIM, 1-st Institutsky proezd, 5, Moscow, 109428, Russia.
[2]Hanoi University of Agriculture, Hanoi, Vietnam

Abstract. Now days automatization of agriculture is perspective and actively developing branch of researches. On this moment there are many international researches which offers ready made solutions for automatic control of sowing agricultures crops. However Most part of sowing are made on a mechanization devices with obligatory control of high qualified service employees in Russia. In this case, it was proposed to automatization of the process of sowing seeds.

1 Introduction

Mechanization and automatization in the agricultural sector can improve productivity and quality of labor. Applied new technologies increase the level of production and the quality of agricultural products. Such processes are directly related to the application of industrial technologies in this industry, as well as to the improvement of planning and management. In agriculture, industrial automation systems used more often. Introduced into the industry electronic innovations can significantly increase production capacity, sales volumes.

The first logical controllers appeared in the form of boxes with a set of interconnected relays and contacts. The scheme was set rigidly at the design stage and could not be changed further.

In the first software-logic controllers (SLC) the operation logic was programmed by the LD connection diagram (line diagrams). These PLS came to replace relay logic controllers. The device had the same operation principle, but the relays and contacts (except for input and output) were virtual, this means that, it existed in the form of a program executed by the SLC microcontroller.

Modern SLCs are free programmable, but it has high cost. It is a need to reduce costs with constant reliability parameters of visualization tools.

The integration of the SLCs and the high-precision units into a seeders allows to improve the quality of sowing, exactly, such parameters as equable distribution of seeds in a row, reducing seeds, reducing the labour intensity of the process, corrective the seed rate.

We are studying the integration of an automated seeding system performed on a single-board Raspberry Pi 3 computer based on the Raspbian embedded system in this article.

The research aim is integration of an automatic sowing system on a single-board Raspberry Pi 3 computer for the selections seeder.

---

[*] Corresponding author: litvvinov.max@yandex.ru

Actuality of this research is increasing productivity by automating the process of sowing seeds, and reducing the **cost of the project**. The sowing process control algorithm is original and its adapted for various kinds of crops. These factors makes it universal.

The market of automatic seeding systems hasn't developed for now days. This system has been already developed for the similar foreign machines such producers as Amazon (Germany), Zürn (Germany), Klen(Russia), Omsk Experimental Plant (OEP)(Russia), Wintersteiger (Austria).

But these machines have the following disadvantages: high price, lack of adaptation to the soil and climatic conditions of Russian Federation, in addition, the used analogue sowing systems monitor the sowing process and signal the operator about problems, and don't correct the sowing rate in accordance with the operating conditions. Therefore, the developed system is more relevant and perspective for future using.

Another positive fact is that foreign manufacturers mainly producer made machines flat-to-flat, i.e. for each customer according to their specific parameters.

2   Materials and methods

The sequence of the signals has been designed in the instrumental software package of industrial automation called CODESYS. The base of this complex is the application development environment for programmable logic controllers (PLCs) [1].

Five program languages are available for programming which defined by the IEC 61131-3:2013 «Programmable controllers - Part 3: Programming languages».

In this case, the program was implemented in the CFC language . This language is addition to the FBD language (function blocks), the basis of which is the arbitrary placement of blocks and the arrangement of the order of their execution.

There were described in detail the hardware of the sowing device In the paper[1] about designing of a microprocessor control system for sowing seeds. It should notice that the SMSD-4.2RS-485 driver is used to control the process of sowing the seeds with a NEMA34 stepping bipolar engine [2].

The source of signal sequence generation is the Raspberry Pi 3 microcomputer. The necessary outputs of the STEP, DIR, GND, and EN driver are connected to the microcomputer via a logic level converter (Fig. 1), since the logical unit of the SMD-4.2 [3] driver is 5 V and the logic unit of the Raspberry Pi 3 is 3.3 V.
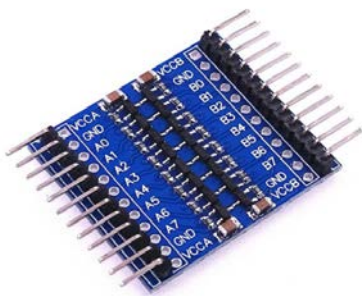
Fig. 1. Signal level converter.

Wintersteiger Seeder (type Rowseed S) was chosen as a prototype for modernization, because the manufacturer provided for the seeder drive from the seeder driven wheel, and the seed rate is changed due to changes in the gear ratio of the gear pairs of the reduction gearbox.

The installation of an automated seeding system will improve quality of sowing through the use of modern precision drives based on stepper engine, as well as abandon the metal-consuming design of the reduction gearbox and chain drive.

3 Results and discussion

It showed the mapping of inputs / outputs of the SMD-4.2 driver to the Raspberry Pi 3 in Table 1.

Table 1. Input / Output Correlation

SMD-4.2 Driver Output	Raspberry Pi 3 Contacts
5v	Pin 2 - 3.3V Power
GND	Pin 6 - GND
EN	GPIO4
DIR-	GPIO27
DIR+ и STEP+	GPIO17
STEP-	GPIO22

A view of the program for controlling the rotation of a stepper engine (SE) in the CODESYS environment is shown in Figure 2 [4].
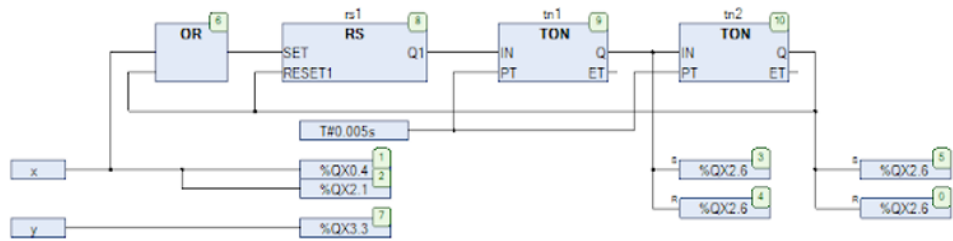

Fig. 2. Control program of stepper engine (SE)

In this program, control is carried out using two buttons "x" and "y", presented in Figure 3.

The "x" button is used to manually start the engine rotation process, while the "y" button is used to switch the direction of rotation of the engine (reverse).

Fig. 3. Visualization of the SE management project.

After ending of the writing the program, important part is the need to switch the connected outputs to the Output state (GPIO4, GPIO17, GPIO22, GPIO27) in the GPIO's tab of the configuration shown in Table 2.

Table 2. GPIO's configuration.

| Parameter	Type	Value	Default value	Description
GPIO4	Enumeration of BYTE	Output	not used	Configuration of GPIO4
GPIO5	Enumeration of BYTE	not used	not used	Configuration of GPIO5
GPIO6	Enumeration of BYTE	not used	not used	Configuration of GPIO6
GPIO12	Enumeration of BYTE	not used	not used	Configuration of GPIO12
GPIO13	Enumeration of BYTE	not used	not used	Configuration of GPIO13
GPIO16	Enumeration of BYTE	not used	not used	Configuration of GPIO16
GPIO17	Enumeration of BYTE	Output	not used	Configuration of GPIO17
GPIO18	Enumeration of BYTE	not used	not used	Configuration of GPIO18
GPIO19	Enumeration of BYTE	not used	not used	Configuration of GPIO19
GPIO20	Enumeration of BYTE	not used	not used	Configuration of GPIO20
GPIO21	Enumeration of BYTE	not used	not used	Configuration of GPIO21
GPIO22	Enumeration of BYTE	Output	not used	Configuration of GPIO22
GPIO23	Enumeration of BYTE	not used	not used	Configuration of GPIO23
GPIO24	Enumeration of BYTE	not used	not used	Configuration of GPIO24
GPIO25	Enumeration of BYTE	not used	not used	Configuration of GPIO25
GPIO26	Enumeration of BYTE	not used	not used	Configuration of GPIO26
GPIO27	Enumeration of BYTE	Output	not used	Configuration of GPIO27

The next step is to control the rotation of the stepper engine with using a laser and a photo sensor.

The laser has 2 wires: power and «ground». The photo sensor has three wires: power, «ground», and a signal. Signal is connected to the GPIO18 input of the microcomputer via a pulse converter (Table 3), as an Input signal which is configured in the GPIO's configuration (Fig.4).

Table 3. Photo Sensor Connection

Signal wire from photo sensor	Contacts of Raspberry Pi 3
A4 on a level converter (for the first laser)	GPIO18
A_ on a level converter (for the second laser)	GPIO24

After we have added the necessary input signal, the control program will be as shown in Figure 4.
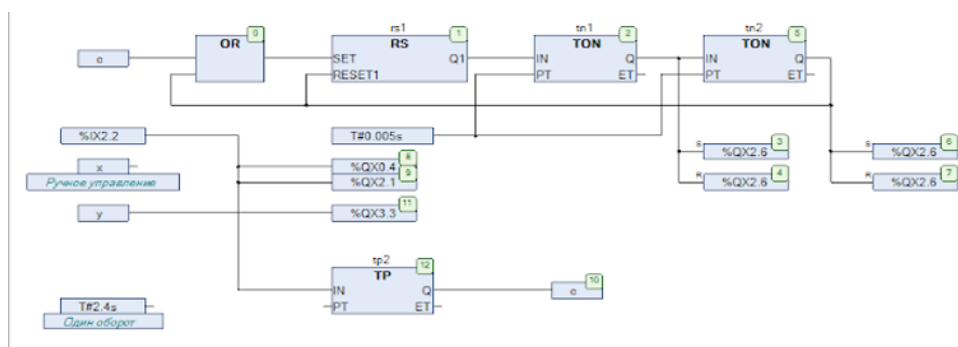
**Fig. 4.** Control program of SE with help the laser sensor

Now, instead of the manual control button (variable "x"), the signal comes directly from the photodiode.

The TP control unit (timer) serves to supply a signal to the SE for the indicated time. It is necessary 2.4 second to make one full turn of the SE in full step mode, therefore, to solve this problem, if its necessary, we should sent the signal to the input of the PT block TP. Signal must be equal to T # 2.4s.

In our case, the signal isn't sent to the PT input, as control is not carried out in time. Control is carried out by the number of steps, i.e. one. This engine has 200 steps per one turn, which means that one step has 1.8 degrees of turn.

The Hall transducer Module RKP-HMS-3144E (Fig. 5) is designed to stop the rotation of a DC motor, which operates as a distribution table drive.

The DC motor is controlled by the VNH2SP30 driver (Fig. 6).
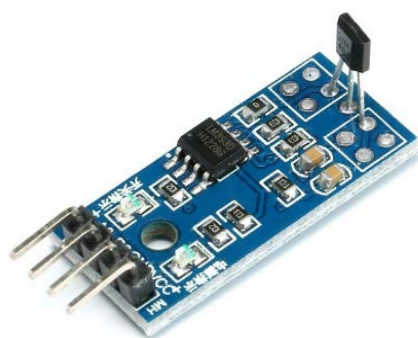
Fig. 5. Hall transducer Module

The module is connected via three contacts (VCC, AO, GND). The signal pin is the analog AO pin which sent to the Raspberry. It connects to GPIO23 via a pulse converter (Table 4) as an input signal, which is configured in the GPIO's configuration (Figure 4).

Table 4. Connection of the Hall transducer module

| Signal wire from Hall transducer	Contacts Raspberry Pi 3
AO	GPIO23

Fig. 6. Collector Motor Driver VNH2SP30

Label on the back of the board:
- +5V — Power 5 Volts, it is not clear why, apparently needed for pull-up resistors.
- GND — No explanation required, mass.
- EN — Enabling the driver To enable the driver, you need to submit a logical unit.
- CS — Current sensor output.
- INA — Driver Input A.
- INB — Driver input B.
- PWM — Input for controlling engine speed.

The connection of the outputs of the PT motor driver is performed in accordance with table 5 [5].

Table 5. Correspondence of driver contacts to Raspberry Pi.

Driver Inputs for PT Engine	Contacts Raspberry Pi 3
+5v	Pin 2 - 5V Power
GND	Pin 30 - GND
EN	GPIO12
INA	GPIO16
INB	GPIO20
PMW	GPIO21

After setting up the software for the automated seeding system on the experimental model, a working version was made which consists of two parts: power (Figure 7), and logical (Figure 8).
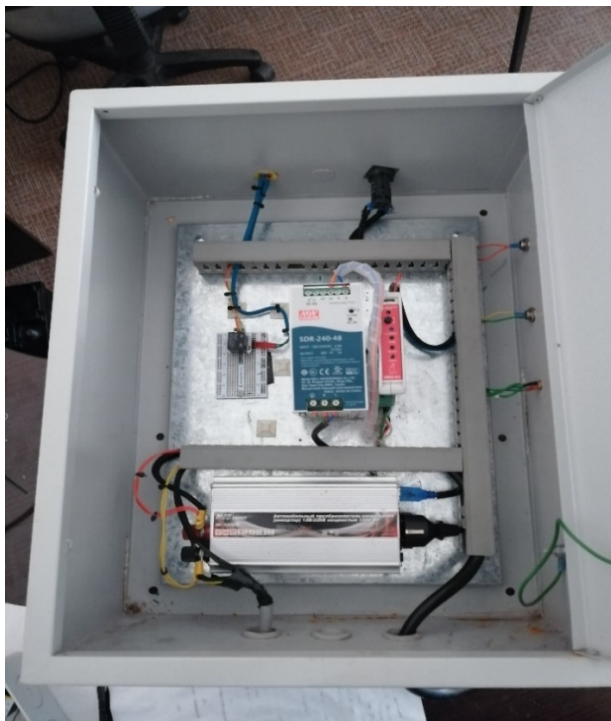
Fig. 7. Power section of an automated seeding system.

The power part includes an inverter 12-220 V for powering the entire system from the battery, driver power supply unit SMD-4.2, driver SMD-4.2, relay for activating the metering solenoid.

Fig. 8. The logical part of an automated seeding system with controls.

The logical part is made in an installation box made of PVC plastic with protection class IP 86. The control interface of the automated seeding system is displayed on a 4.7-inch touch screen. It is also possible to remotely connect to the system through a telephone or computer via a WI-FI network.

The parts of the automated seeding system are connected to the units of the seeder by means of pin connectors and plugs with fixation by a threaded connection.

4 Conclusions

In the course of the above operation, a seed sowing operating program was designed, based on which:
-step engine carries out the process of scattering of seeds,
- a laser sensor and a photodiode control the rotation speed of the tractor wheel, and set the rotation speed of the stepper motor,
- Hall transducer controls the process of turning on / off the solenoid,
-solenoid provides grain feed to the stepper motor,
- DC engine is used for feeding / moving seeds (cassettes) along the conveyor.
The hardware for the seeding control system was also designed and manufactured.

References

1. CODESYS Development System V3 // https://store.codesys.com URL: https://store.codesys.com/codesys.html#Produktbeschreibung. (date of request: 21.02.2020).
2. Litvinov M.A., Adamian A.A., Moskovsky M.N., Development of a microprocessor control system sowing seed. Engineering journal of Don. № 7, p 3. (2019).
3. The stepper motor controller SMSD-4.2 with RS-232 and RS-485 **//** electroprivod.ru URL: electroprivod.ru/smsd-42-rs232-rs485.htm. (date of request: 14.03.2020).
4. Adamian A.A., Al-Tibbi V.H., Stepper motor control based on the single-Board computer raspberry Pi 2 in the environment CODESYS V3.5. Young researcher of the Don № 4(13), p 12. (2018).
5. Petrov I.V., Dyakonov V.P., Programmable controller. Standard languages and techniques for applied design, Moscow: SOLON–Press, 256 p. (2004).