

PAPER • OPEN ACCESS

## CodeSys based HMI design and sensor upgrades for a PLC and Atmega2560 controlled industrial conveyor

To cite this article: R Varga *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **568** 012062

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the [collection](#) - download the first chapter of every title for free.

# CodeSys based HMI design and sensor upgrades for a PLC and Atmega2560 controlled industrial conveyor

**R Varga, T I Erdei, G Husi**

Department of Mechatronics Engineering, University of Debrecen, Faculty of Engineering, Debrecen, Hungary

vargarul94@gmail.com

**Abstract.** Engineers nowadays must be able to solve the challenges set by industry 4.0, in order to make existing devices viable for this new trend. This requires a complex, system-based way of thinking. Nowadays, in most cases, we usually find company-specific hardware and software in industrial automated production. However, in different industries, it is not unheard of that a given manufacturing cell, robot, or manipulator became unserviceable, because a manufacturer is no longer on the market, making it impossible to find replacement controllers, software, or updated firmware for our industrial robot, or manipulator. The following project aims to provide a solution to the problem detailed above, where an unserviceable Flexlink conveyor's controller is replaced with both an Arduino and a PLC, and before realisation of the new control method, simulations are possible before connecting the actual controller.

## 1. Introduction

In the building-mechatronics research centre of University of Debrecen, a new lab has been created with the name of „Cyber-Physical and Intelligent Robot Systems Laboratory”. The goal of this laboratory is to design and apply new technologies in the field of robotics, which are later to be integrated into other fields as well. In our robotics lab, numerous upgrades and research are underway, which include the designing of a new conveyor belt system.

The functions of the KUKA KR5 found at the Department of mechatronics [1] have been extended during previous upgrades, namely the robot cell built around it. One of the key components of this cell is the conveyor belt. During its conversion, one of the main focus was to ensure simple connection and usage of various controllers, and to choose controllers which can be replaced in case of their malfunction, without having to modify the existing system. By this, we can avoid the situation where a manufacturer disappears, as the shown example in the abstract. Two particular controllers were chosen with specific properties: one of them is open-source, while the other conforms industrial needs and conditions. In both cases, the existence of a simulation environment is crucial.

## 2. ATmega2560 & 328 microcontroller

In order to ensure the open-source nature of the controller, and to guarantee its usability in an educational setting, the Arduino product line was chosen, which also has extensive support.

The Arduino Mega 2560 Rev3. has 54 digital I/O ports, 16 analogue inputs, 4 UART communications port, a 16MHz crystal oscillator, and an USB connector, power supply connector, an ISCP header, and a “Reset” button. This version of the Arduino serves the role of a “pre-tester” device in the control of the conveyor belt.

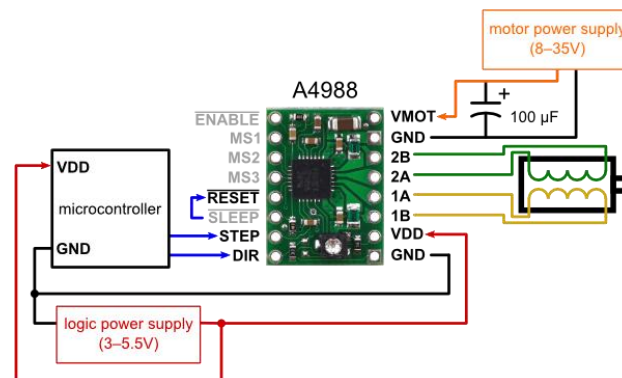


In terms of programming, the Arduino IDE development environment is available, which is a cross-platform, java-based tool. In terms of Arduino, another model was also chosen, which is the Arduino Nano. This device has been added to the system not with a pre-testing consideration, but as a device to connect the PLC to the conveyor belt, with the goal of stepper motor control. The panel's main component is the ATmega328 microcontroller, which is available with 32KB of internal flash memory, 14 digital I/O pins and 8 analogue inputs and works at a frequency of 16 MHz. Its operational voltage is 5V, although the input voltage may vary between 7 and 12V.

### 3. NEMA23 stepper motor and Pololu A4988

To drive the conveyor belt, a NEMA23 Nanotec Munich made ST5818L type stepper motor was used, with a step angle of 1,8 degrees, which can be operated in both bipolar, and unipolar modes [9].

For the control of the stepper motor, a stable, reliable motor controller was required. The chosen controller was the Pololu A4988, which sees wide-spread use in 3D printers, and CNC machines [10]. The five different stepping resolutions are set by MS1, MS2, MS3 inputs. The MS1 and MS3 have a pull-down resistor of 100k $\Omega$ , while the MS2 has a 50k $\Omega$  one [2].



**Figure 1.** Pololu A4988 stepper motor controller [2]

### 4. PLC & conveyor belt

For the PLC itself, a modular Schneider Electric Modicon M340 [3] was chosen, as it's easily customisable. In the project, the to-be controlled conveyor belt is found in the robotics lab, which is part of the FlexLink XK product line used in car and food industries. This particular FlexLink conveyor belt was chosen, because there are no plans of moving workpieces during education, that are greater than 10[kg] in weight. The total maximum weight the conveyor can carry is 30[kg]. The plastic belt is 45[mm] wide, and 5200[mm] long [4]. Two XUB0BPSNM12 photoelectric sensors were secured on the conveyor, which are part of the OsiSense XU product line [5].

### 5. ATmega2560 pre-tester

Assembly of the conveyor belts and its stepper motor was done in an earlier project in the robotics lab. However, the original FlexLink controller is non-functional, and the original development environment is only compatible with the specific controller. As a result, without replacing the original hardware, the conveyor could not be operated.



**Figure 2.** Conveyor belt with sensors

In order to re-integrate the conveyor into future projects and education, the original hardware had to be replaced. The Arduino Mega2560 was a precursor to the final controller. As it was shown earlier, two photoelectric sensors were fastened to the frame of the conveyor belt. Four terminal blocks were also secured on this frame, assigned to negative-positive, and A-B allocation.

The terminal blocks marked by “A” is connected to the sensor placed at the front part of the conveyor, while “B” is connected to the sensor on the other end. The usage of terminal blocks was warranted by the nature of PLC connections: the screw terminals of both the PLC and the terminal blocks ensure quick connection of wires to the desired devices.

The perforated sheet, serving as a mounting point for the “PLC controller wall” also holds the five button Schneider button box. A total of five inputs were connected to the PLC’s input module. With these, “manual” control of the conveyor belt can be achieved (start/stop).

The “PLC controller wall” also contains another Schneider box, housing five red LEDs indicator lights, which are connected to the PLC’s output module. These serve indication purposes and provide visible feedback about the program’s state.

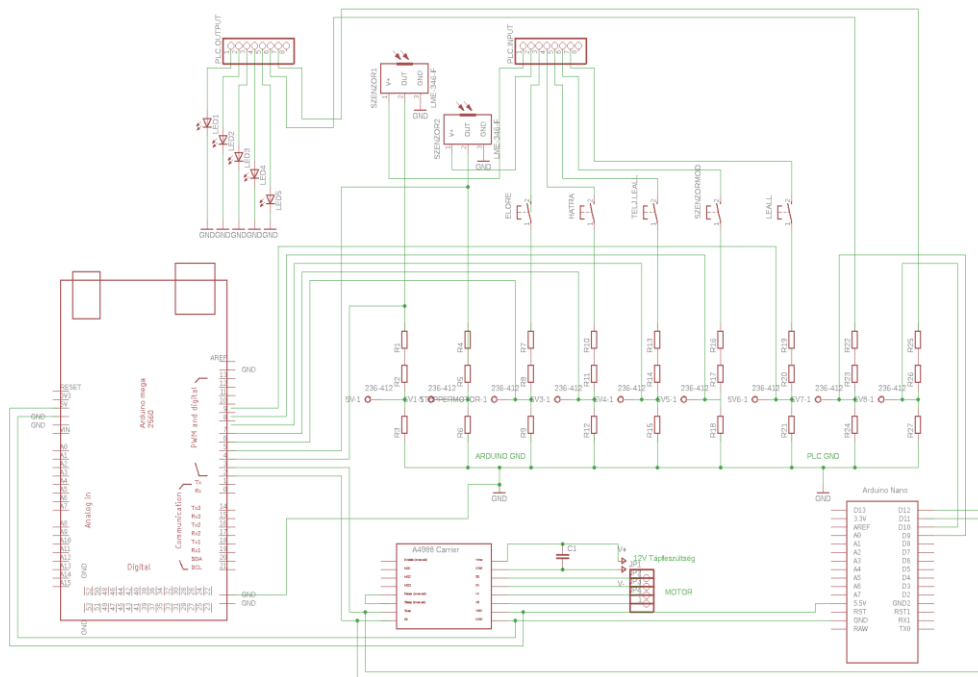
The Arduino ATmega 2560 was also secured on this panel, including the breadboard housing the necessary electrical components. A 3D printed, DIN rail compatible adapter made of PRO PLA was used as mounting point. The Arduino Nano is mounted in a similar fashion.

During pre-testing, an Arduino ATmega2560 microcontroller and a suitable stepper motor controller, A4988 was used to drive the stepper motor itself, the Nanotec Munich ST5818L.

The external 12V power supply was an AX-3003D bench power supply [6]. This was required because the 3.3V/5V output level of the Arduino was insufficient, and also because the rated maximum current output of the ports is also only 40[mA], which is not enough for the used NEMA23 stepper motors.

As the five pushbuttons and the two sensors both require 24 V voltage, a voltage divider was required to connect them to the Arduino board.

After every device necessary for the electrical control was prepared, it was necessary to ensure their connection. This can be observed on the schematic drawing, seen on figure 8.



**Figure 3.** Circuit schematics

Every pushbutton received its own function. As such, pressing the button connected to pin 6, the conveyor belt automatically moves forward. Pressing the button connected to pin 8 stops the conveyor. Only after this can the button connected to pin 7 be activated, to start the conveyor to move in the opposite direction. This serves as a safety function, to avoid motor overload caused by the direction reversal. Sensor mode is activated, when the pushbutton connected to pin 9 is activated. This mode can be deactivated by pushing the button connected to pin 10, which deactivates the conveyor belt, and causes it to stop.

After finishing the program, it was verified by the Arduino IDE, and uploaded to the Arduino Mega 2560 board. Afterwards, the setup was tested within the KUKA KR5's work zone.

## 6. CodeSys PLC and HMI

If we want to apply the previously written and tested program onto the final, industrial hardware, there's one serious limitation to consider.

The Arduino IDE provides means to check our written program, however, if we'd like to transfer this same program to PLC, we might not be given the same opportunity of troubleshooting. One of the reasons for this, is the large number manufacturer-specific PLC programming environment, which often only allow simulation through their own, separate "plugin", to check the program's viability [7]. To combat this, a software called "CodeSys" was used, which is free, and its features include PLC programming, simulation, and HMI (human machine interface) design [8].

During the program transfer from Arduino to PLC, first the previously detailed buttons were designed as part of the HMI. After designing the HMI, the Arduino program was written in PLC Ladder, and the HMI's buttons were assigned.

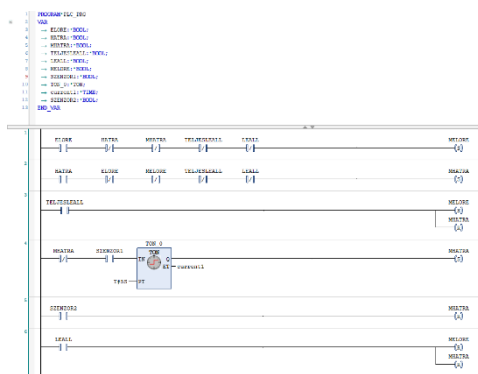


Figure 4. CodeSys – PLC program

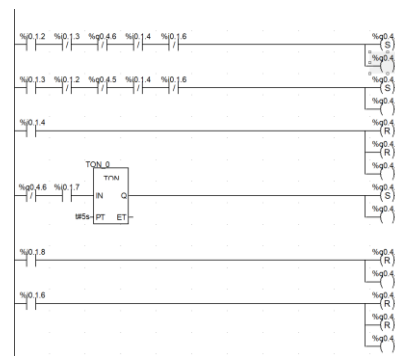


Figure 5. Unity - Ladder diagram program

Later the program written in CodeSys was ran, to ensure its error-free function. Another positive effect of this, is that a PLC based conveyor model was gained, which may be used in future controllers or HMI projects, as it is freely editable.

## 7. Industrial PLC finalising

After the previously written CodeSys program was checked, the already simulated PLC Lad program was copied to the Schneider Electric Unity development environment. The timer also only functions with a 5 seconds delay. The physical input addresses range from the second digital input to the eighth and are found on the first module. They are addressed in the following form: “%i.0.0.2”. The physical outputs range from the zeroth digital output to the sixth and are located on the fourth module. They are addressed in a similar manner as inputs are (as an example: “%q0.4.0”). Timer values are also 5 seconds.

The sensors and pushbuttons were connected to the input module, while the indicators and the outputs necessary for the stepper motor's function were connected to the output module. In this program, lamps had to be set as normal coils (outputs), as they are responsible in providing feedback about the program's state. The connection between the computer and the PLC are establish via a UTP cable, and a router. Within Unity, the PLC's IP address must be defined, to guarantee stable communication with the Schneider Electric Modicon M340 PLC.

At the moment of controlling the conveyor belt, the PLC was equipped with BXM DDI 1602 input modules, and BXM DDO 1602 output modules. Because these modules are incapable of generating the high frequency signal necessary for controlling the stepper motors, the previously mentioned Arduni Nano board was chosen to generate signals to be used by the Pololu A4988 motor controller. The Ardunio, depending on its inputs, either sets the conveyor's rotation direction, or stops it. The angular speed of the NEMA23 is constant. As the Ardunio Nano operates on a 5V voltage level, the 24V signal coming from the PLC had to be connected to a voltage divider first.

## 8. Conclusion

The conveyor belt found in the „Cyber-Physical and Intelligent Robot Systems Laboratory” had its missing controller successfully replaced, which paves the way for the usage of new sensors, and controller shields.

In terms of hardware, two controllers are used: Ardunio fulfilling a “pre-tester” function, and the industrial PLC, serving as the final controller, using the results acquired by during pre-testing.

Simulation is possible in either cases, to check the created program's reliable operation, before testing with real hardware.

It can be said, that a cost. effective, flexible system was created, with easy-to-replace components. In addition, a previously inactive assembly line device was restored to working condition, which now works reliably.

## Acknowledgement

Hereby I'd like to express my thanks towards Dr. habil Husi Géza and Erdei Timotei István, and also towards the „Cyber-Physical and Intelligent Robot Systems Laboratory” and its members.

## References

- [1] Erdei T I, Molnár Zs, Obinna N C and Husi G 2016 *Industrial KUKA Robot Manipulator in Simulation Environment and Position Read-back*
- [2] *Pololu A4988 stepper motor controller* [Online]. (2019.01.12). Available: <https://www.pololu.com/product/1182>.
- [3] *Modicon M340* [Online]. (2019.01.12). Available: <https://www.se.com/hu/hu/product-range-presentation/1468-modicon-m340/>
- [4] *Flexlink Conveyor system XK* [Online]. (2019.01.12). Available: [https://www.flexlink.com/zcms/zpublish/45/uploads/45/tech\\_lib/14170936898875465656\\_08XK.pdf](https://www.flexlink.com/zcms/zpublish/45/uploads/45/tech_lib/14170936898875465656_08XK.pdf)
- [5] *XUB0BPSNM* [Online]. (2019.01.12). Available: <https://www.schneider-electric.com/en/product/XUB0BPSNM12/photo-electric-sensor---xub---multi---sn-0..20m---12..24vdc---m12/#search>. Available: 2018.11.09.]
- [6] *AX-3003D power supply* [Online]. (2019.01.05). Available: [https://www.hestore.hu/prod\\_10025553.html](https://www.hestore.hu/prod_10025553.html)
- [7] T I Erdei and Zs Molnár 2018 *PLC – Applied Automation I. (PLC - Alkalmazott automatizálás I.)*
- [8] *CodeSys* [Online]. (2019.01.04). Available: <https://www.codesys.com/>
- [9] *Nanotec catalogue* [Online]. (2019.01.08). Available: [https://en.nanotec.com/fileadmin/files/Kataloge\\_Flyer\\_Archiv/Nanotec\\_Catalog\\_2007.pdf](https://en.nanotec.com/fileadmin/files/Kataloge_Flyer_Archiv/Nanotec_Catalog_2007.pdf)
- [10] Erdei T I, Molnár Zs and G Husi 2016 *Self-reproduction 3D printer in controlled environment* InterTalent UNIDEB - University of Debrecen International Conference