POLITECNICO DI TORINO

Master’s Degree in Mechatronics Engineering

Master’s Degree Thesis

Connection between Raspberry PI and Codesys software and their implementation in an electro-pneumatic test bench

Supervisor

Prof. Luigi Mazza

Candidate

Bekhzod Sobirov

July 2021

Abstract

With the advancement and transformation of a technology over the past decades the industrial structure has changed substantially. Hence, the Industry 4.0 transformed from the conventional into the intelligent one. This kind of change has been possible due to the implementation of new technologies and techniques.

Speaking of new technologies, IoT (Internet of things) has massively impacted onto the development of Industry 4.0. IoT is a rapidly growing technology whose main aim is to connect physical world to the digital one and allow people and “things” to be connected anytime, anywhere ideally using any network and service.

There are different kinds of technologies in IoT, but this thesis covers only Raspberry Pi board. The main aim of this thesis work is to show the cooperation of Raspberry Pi board and Codesys software and their implementation in control of electric and pneumatic components in a test bench. Moreover, the thesis work covers the software distance control approach of the electric and pneumatic components.

Keywords: Raspberry Pi; Codesys; electro-pneumatic components.

Content

List of figures **1**

1 Introduction **3**

2. CAD model of electro-pneumatic test bench and hardware construction .. **5**

2.1 Test bench model in Solidworks 5

2.2 Construction of hardware 6

2.2.1 DC motor 6

2.2.2 Push-pull solenoid 8

2.2.3 Inductive and capacitive sensors 10

2.2.4 Electro-pneumatic cylinder 13

3. Software design **15**

3.1 Connection between Raspberry Pi and Codesys software 15

3.2 Implementation of components in Codesys 24

3.2.1 DC motor 24

3.2.2 Inductive and capacitive sensor 28

3.2.3 Push-Pull solenoid 30

3.2.4 Electro-pneumatic cylinder 30

3.2.5 Raspberry Pi camera 31

4. Conclusion and future perspectives **34**

5. Bibliography **35**

LIST OF FIGURES

Figure 1: CAD model of the test bench.

Figure 2: CAD model of test bench (side view).

Figure 3: Electric schematics to control DC motor.

Figure 4: Electric schematics to control push-pull solenoid.

Figure 5: Electric schematics to control motor and push-pull solenoid.

Figure 6: Electric schematics for inductive or capacitive sensor.

Figure 7: Working principle of optocoupler.

Figure 8: Signal conversion of inductive and capacitive sensors.

Figure 10: Setting up parameters for sd card formatter.

Figure 11: Completion of formatting by sd card formatter.

Figure 12: Special purposed installer of noobs.

Figure 13: Website of the noobs operational system.

Figure 14: Illustration of complete of flashing.

Figure 15: Creation of additional files in sd card.

Figure 16: Changing the extension of the wpa_supplicant.conf.

Figure 17: Pinging computer with Raspberry Pi.

Figure 18: Pinging performed successfully.

Figure 19: Web site for downloading library of Raspberry Pi.

Figure 20: Installation of library.

Figure 21: Installation of library.

Figure 22: Setting up connection parameters.

Figure 23: Blink function block.

Figure 24: Control panel.

Figure 25: Device log.

Figure 26: Assignment of GPIO pin of motor as output.

Figure 27: GPIO pin for motor.

Figure 28: Assignation of inductive and capacitive sensors into GPIO.

Figure 29: Assignation of the inductive sensor signal in control panel.

Figure 30: Assignation of the capacitive sensor signal in control panel.

Figure 31: Assignment of solenoid control variable.

Figure 32: Assignment of the variable to control movement of cylinder piston.

Figure 33L Web browser in the visualization.

INTRODUCTION

Raspberry Pi boards are special designed computers which were designed as initial step as educational computers to get acquainted with special purposed applications.

But nowadays, the application of the Raspberry Pi boards goes beyond the educational purpose. It is possible to apply Raspberry Pi boards even in industrial applications. The examples could the control of the industrial components like sensors, actuators and others.

Moreover, the Raspberry Pi boards can be seen as embedded computers, just because the evaluation of the Raspberry Pi boards is increasing day by day and this is leading to the application of Raspberry Pi boards in different fields.

This thesis work only covers some of the aspects of the application of the Raspberry Pi boards.

Chapter 2.

CAD MODEL OF ELECTRO-PNEUMATIC TEST BENCH AND HARDWARE CONSTRUCTION

2.1 TEST BENCH MODEL IN SOLIDWORKS

The construction of hardware of the test bench starts with the computer aided design model of the test bench plant. The test bench plant consists of 12V DC motor, 1 push pull solenoid, 1 Inductive sensor, 1 capacitive sensor and 1 pneumatic cylinder. Down below given the mechanical construction of the test bench plant in Solidworks software (figure1 and figure 2).

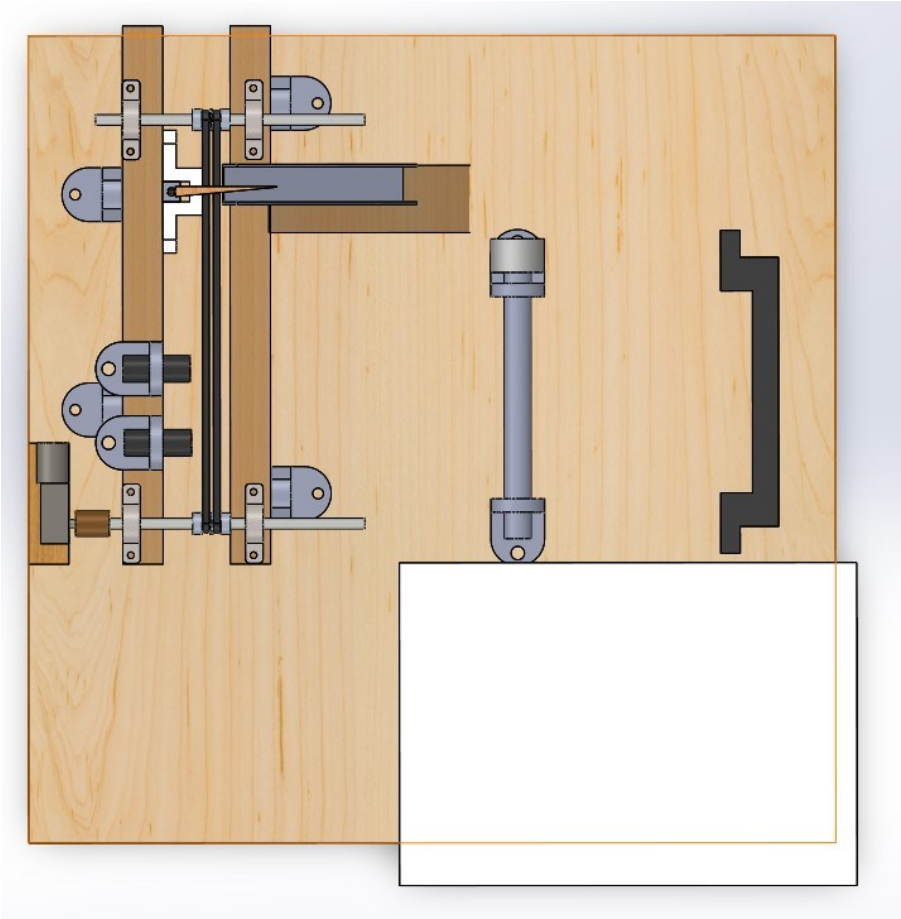

Figure 1. CAD model of the test bench.

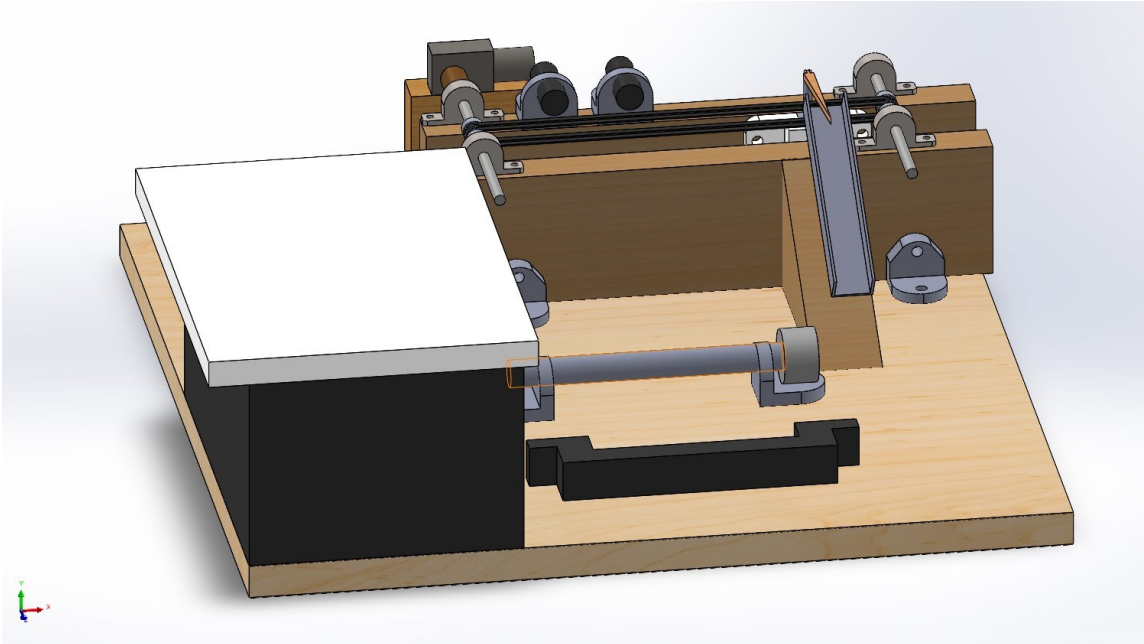

Figure 2. CAD model of test bench (side view).

2.2 CONSTRUCTION OF HARDWARE

2.2.1 DC MOTOR

For the movement of conveyor belt the simple 12V DC motor was chosen. As it is well known that the pins of the Raspberry pi can only output or input 3.3V. The specific electric hardware must be constructed so that Raspberry pi be able to control with only 3.3V the DC motor. Figure 3 shows the electric schematics with which Raspberry pi is able to control 12V DC motor.

Electric schematics consists of 1k Ohm resistor, 5403 diode, TIP 120 NPN transistor. From the schematics, it is possible to understand that the voltage given by the GPIO pin of Raspberry Pi serves as a switch in the whole electric circuit.

In other words, the DC voltage can surpass through the motor if the given current to the base pin of the transistor is sufficient to open the connection between the collector and base pins of the transistor. If we look at the datasheet of the TIP 120 NPN transistor we can see DC minimum current gain is equal

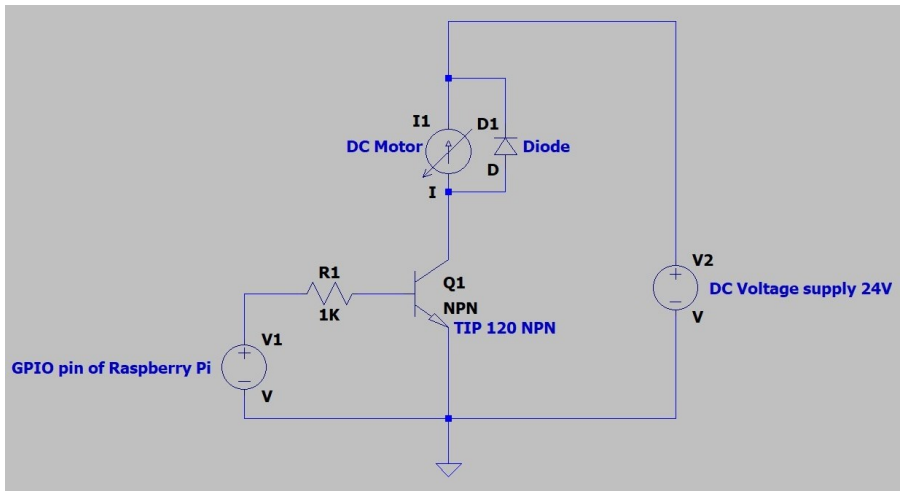

Figure 3. Electric schematics to control DC motor.

to h_{fe} =1000 and we can calculate the minimum base resistor to build our electric schematics from the given formula:

$$I_b=I_c/h_{fe(min)} \tag{2.1}$$

$$R_b=V/I_b \tag{2.2}$$

$$V=V_{bat}-0.7 \tag{2.3}$$

where, I_b is base current, I_c is collector current, R_b is the resistor which is connected to base pin of the transistor and V_{bat} is the DC voltage connected to the collector and emitter pins of the transistor. As it is possible to see from the formula that V is equal to V_{bat} – 0.7. By subtracting 0.7V we are cancelling all kinds of minor errors.

The working principle of the schematics is based on switch circuit. The current from the GPIO pin of the Raspberry pi opens the surpass of the current from voltage supply so that current passes from the collector pin of the transistor to the emitter pin of the transistor, in this sense the closed circuits forms.

2.2.2 PUSH-PULL SOLENOID

The electric schematics for the control of the push-pull solenoid is shown in the figure 4.

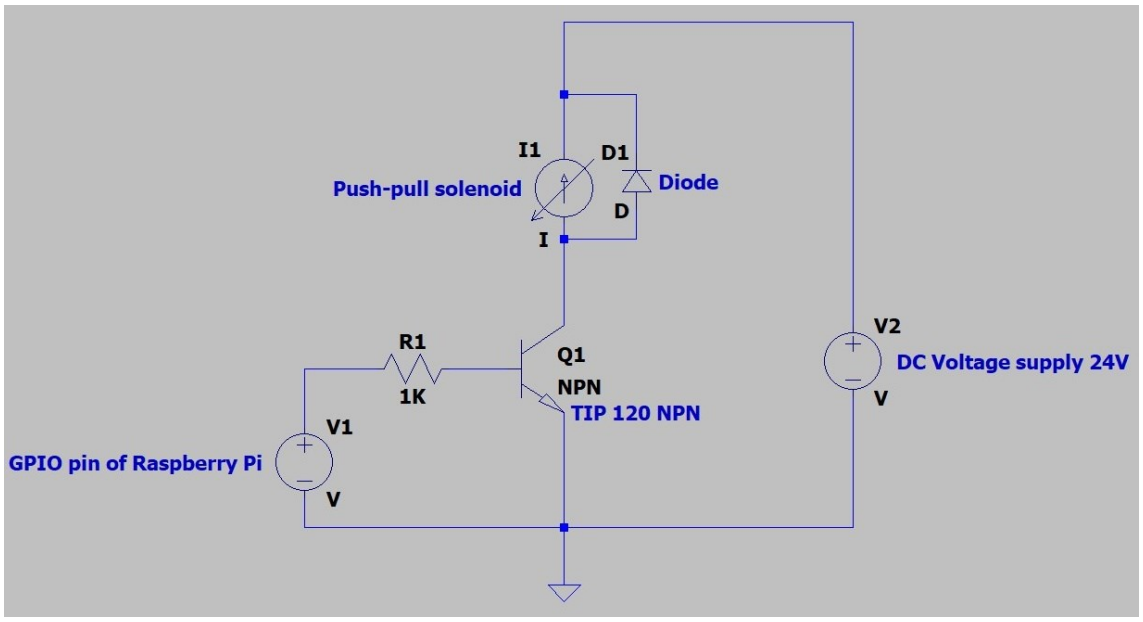

Figure 4 Electric schematics to control push-pull solenoid

The choice of the transistor is arbitrary, but special attention must be paid into the minimum dc current gain, maximum collector-emitter voltage, maximum collector-base voltage, and maximum emitter-base voltage. In the condition above, the choice of the transistor does depend on the 3 factors, i.e., minimum dc current gain, maximum collector-emitter voltage, and maximum emitter-base voltage.

In the condition above, the operational voltage of the push-pull solenoid is 12 V and the maximum collector-emitter voltage that transistor can tolerate is 60V which means that the TIP 120 NPN transistor fits the condition.

As it is well known that the pins of the Raspberry Pi supply voltage 3.3V and this factor was taken into consideration in the choice of the transistor. The transistor TIP120 NPN can bear maximum emitter-base voltage up until 5V and it is sufficient to bear the 3V coming from the GPIO pin of the Raspberry Pi board.

Another quite important factor which lead to the choice of the TIP120 NPN transistor is minimum dc current gain. The minimum dc current gain of the TIP 120 NPN transistor is 1000 and this factor leads to the choice of the resistor which is connected to the base pin of the transistor based on the given below formulae:

The working principle of the electric schematics is based on switch circuit. The current coming from the GPIO pin of the Raspberry Pi turns on or off, i.e., opens or closes the current passing from the collector pin of the transistor to the emitter pin of the transistor. By doing so the part where 24 dc voltage connected to the 2 pins of the transistor forms closed circuit and makes the push-pull solenoid work.

$$I_b = I_c / h_{fe(min)} \tag{2.4}$$

$$R_b = V / I_b \tag{2.5}$$

$$V = V_{bat} - 0.7 \tag{2.6}$$

In the condition above 1KOhm resistor was chosen and this amount of the resistance is sufficient to pull up the required amount of the current from the GPIO pin of the Raspberry Pi to open or close the gate between collector and emitter pins of the transistor.

Moreover it is possible to see in figure 6 the real schematics of the motor control circuit and the push-pull solenoid.

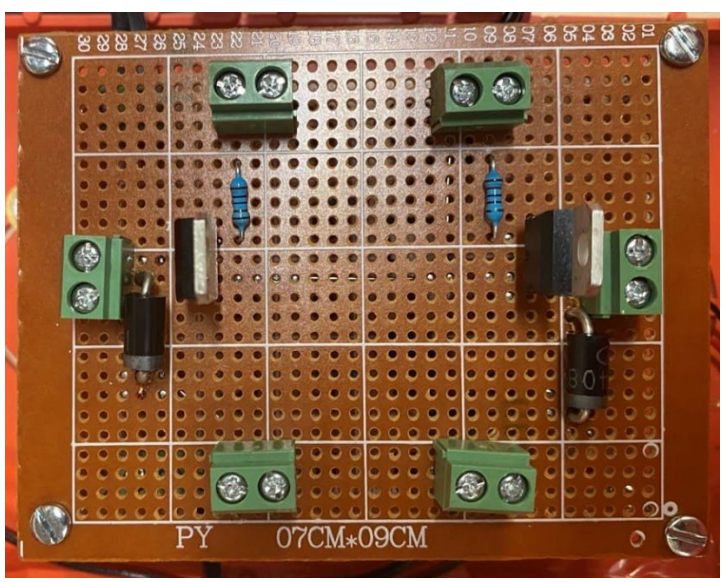

Figure 5. Electric schematics to control motor and push-pull solenoid.

2.2.3 Inductive and Capacitive Sensors

The process of sorting is done by 2 sensors, i.e., inductive, and capacitive sensors. The inductive sensors measure the inductivity of an object which is placed in front of it. In other words, if the object metallic or not.

Whereas the capacitive sensors measure the capacitance of an object placed in front of it, that is, if the object plastic or not.

The choice of the sensors lead to the Festo group sensors. There are multiple reasons for doing so. Which are well function-ability, good sensitivity to measure the object parameters placed in front of them and the last but not least cost effectiveness.

The sensors have 3 pins, i.e., 24V, ground and signal pins. All of the pins work with 24V. This 24V range of the signal pin of the sensors must be changed for the input pin of the Raspberry Pi board. Because the pins of the Raspberry Pi board work only in the range of -3.3V and 3.3V. That is -3.3V is the maximum input voltage and 3.3V is maximum output voltage that Raspberry Pi board can input and supply respectively.

For the conversion of the 24V of the signal pin of the 2 sensors the special designed circuit was made. The conversion is made by the following electric schematics (figure 6).

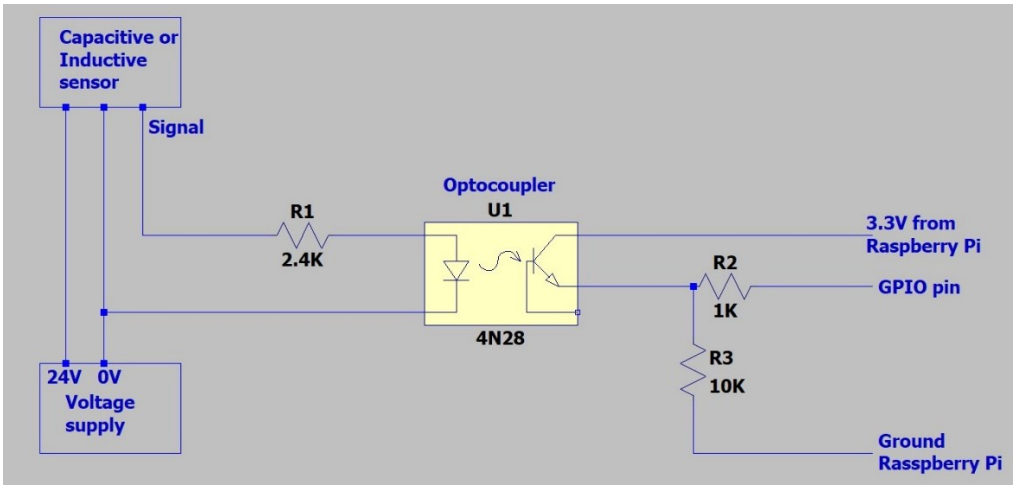

Figure 6. Electric schematics for inductive or capacitive sensor

The figure 7 illustrates the working principle of the optocoupler. A current is first applied to the optocoupler in the left side of the optocoupler, which makes then the infrared LED emit a light that is proportional to the current. When the light hits the photosensitive device, it switches on and starts to conduct a current as any ordinary transistor.

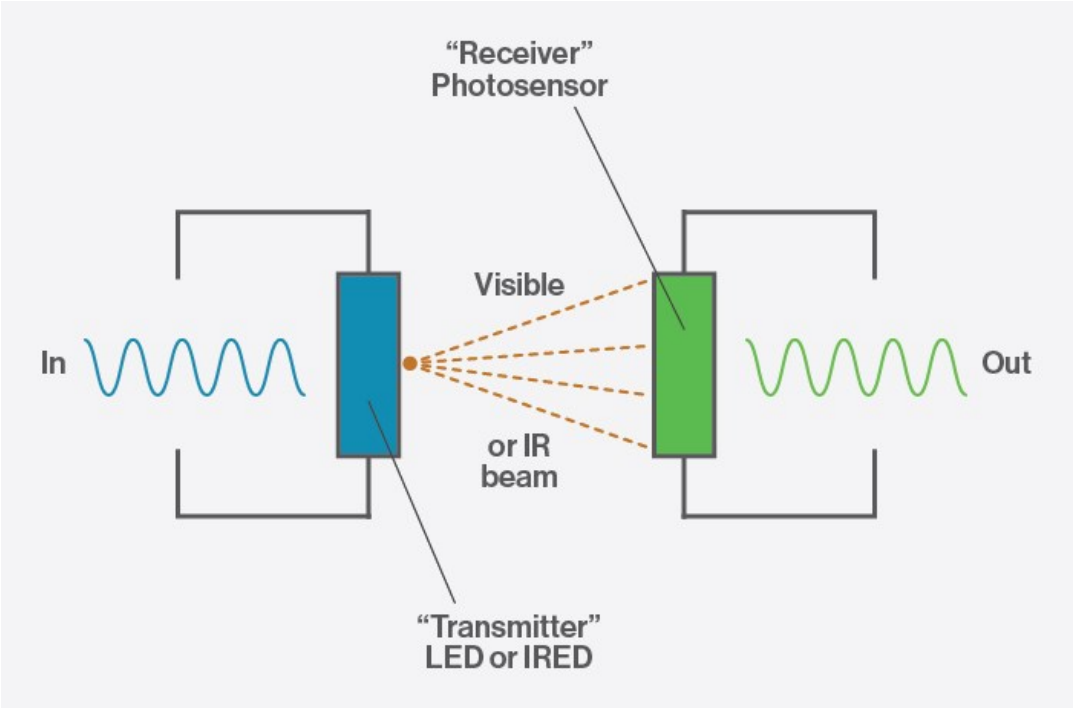

Figure 7. Working principle of optocoupler

Based on the working principal of the optocoupler, the above illustrated circuit for the sensors work almost in the same principle. When there is voltage in the signal pin of the sensor, the IRED or LED emits infrared light, which then effects the receiver, i.e., photosensor to switch on and conduct the current coming from the 3.3V pin of the Raspberry pi board. This 3.3V from optocoupler goes to the ground and GPIO pin of the Raspberry Pi board. The 2 resistors connected to the GPIO and ground pins pull up certain amount of the current from the optocoupler. 1K Ohm and 10K Ohm resistors were chosen for the GPIO and ground pins of the Raspberry Pi respectively.

The amount of the current which surpasses to the pins inversely proportional to the amount of the resistance chosen, i.e. higher for GPIO pin and lower for ground pin of the Raspberry Pi board.

Down below illustrated the real schematics of the conversion of the signal of inductive and capacitive sensors to the GPIO pins of the Raspberry Pi board (figure 8). The consisting components are 2 4N28 optocoupler, 2 2.4kOhm resistors to pull up current from the signal pins of the sensors, 2 1kOhm resistors to pull up current from the 3.3V pin of the raspberry Pi board and 2 10KOhm resistors to pull up current from the 3.3V pin of the Raspberry Pi board.

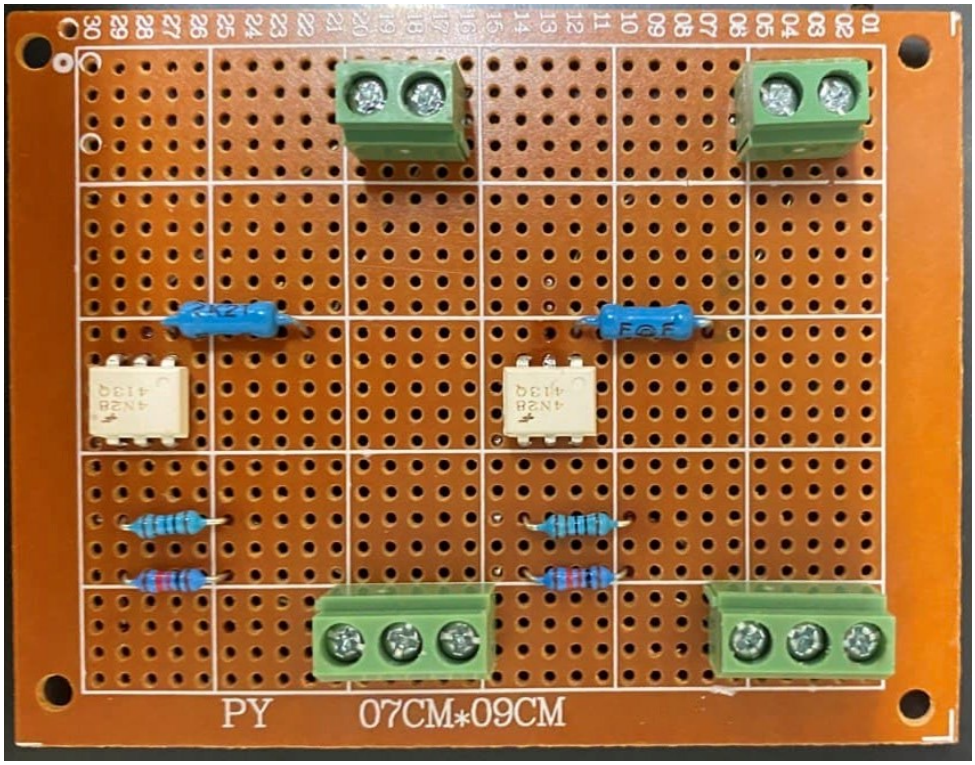

Figure 8. Signal conversion of inductive and capacitive sensors

2.2.4 ELECTRO-PNEUMATIC CYLINDER

For the control of pneumatic cylinders electric valves are usually used. As it is well known that the electric valves usually work in the range of 24 V. In order to control the 24V electric valve the special designed electric schematics must be made. Figure 9 illustrates the electric schematics to control the 2 terminals of the electric valve.

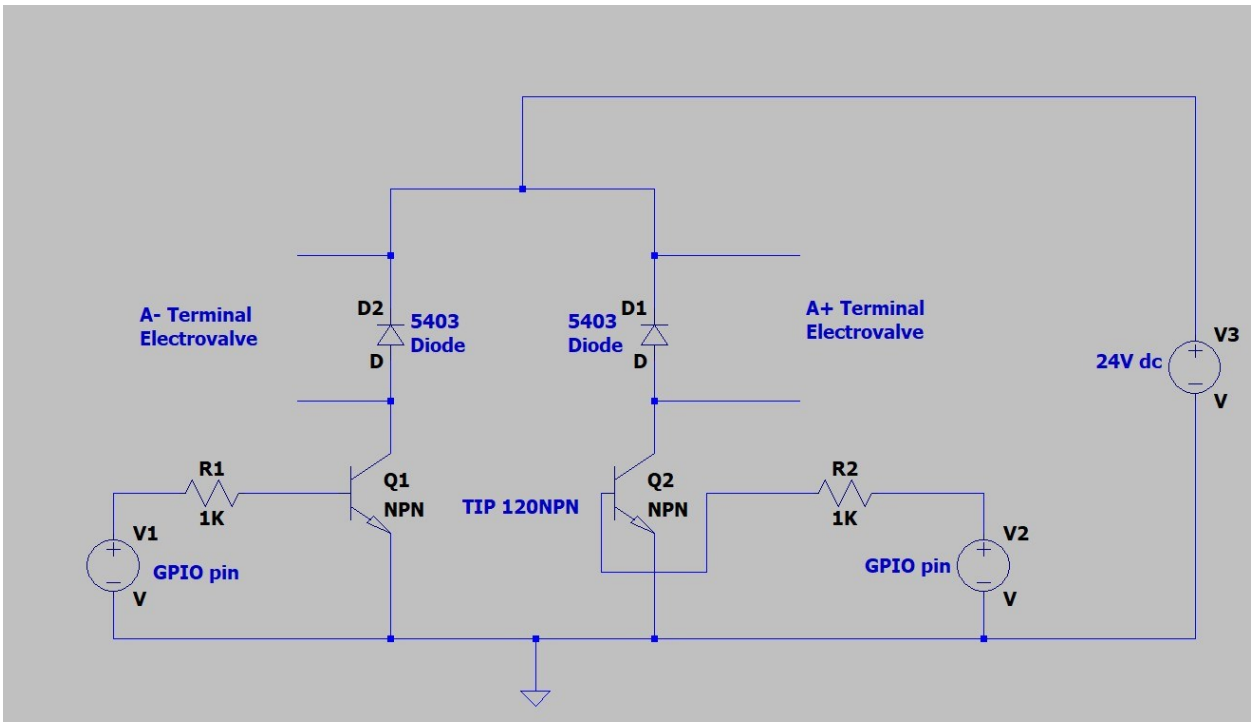

Figure 9. Electric schematics to control cylinder electro-valve.

The supply voltage of 24V dc is going through the collector and emitter pins of the any transistor when the there is enough amount of current applied by the GPIO pin of the Raspberry Pi board.

By correctly applying, i.e., giving enough amount of current to the base pins of the transistor through GPIO pins of the Raspberry Pi I it is possible to control the movement of the pneumatic cylinder.

Chapter 3

SOFTWARE DESIGN

3.1 PREPARATION OF SD CARD FOR THE INSTALLATION OF OPERATIONAL SYSTEM, I.E. NOOBS

As a first step, the sd card must be prepared. There are different options to do the formatting of the sd card, examples could be EaseUs Partition Master, SD card formatter and others. As an example, the sd card formatter has been used.

In order to format the sd card with the help of sd card formatter, the overwrite format option should be chosen. The overwrite formatting is the type of formatting which erases all the hidden auxiliary files and prepares the sd card for installation of the operational system. (figure 10).

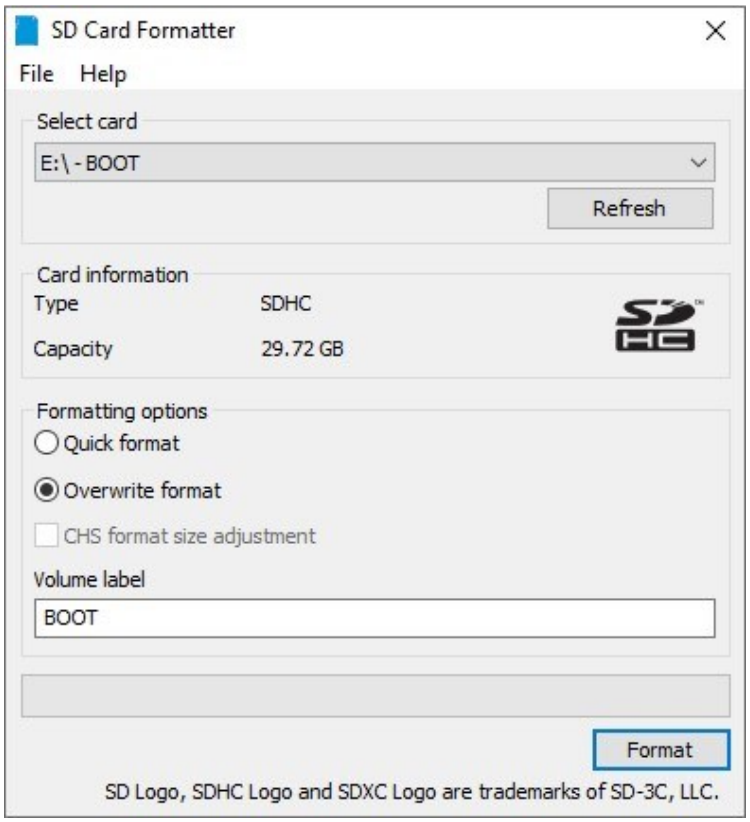

Figure 10. Setting up parameters for sd card formatter

After setting up the correct parameters of the sd card formatter, the format button must be pressed. As the formatting continues some pop-up windows may appear. Therefore, no action must be performed. Because these pop-up windows are process pop-ups and they automatically open and close.

As the formatting finishes, a pop up window will appear indicating for the completion of the process. In this pop-up windows there would be written "Formatting was successfully completed" as it is illustrated in the figure 11.

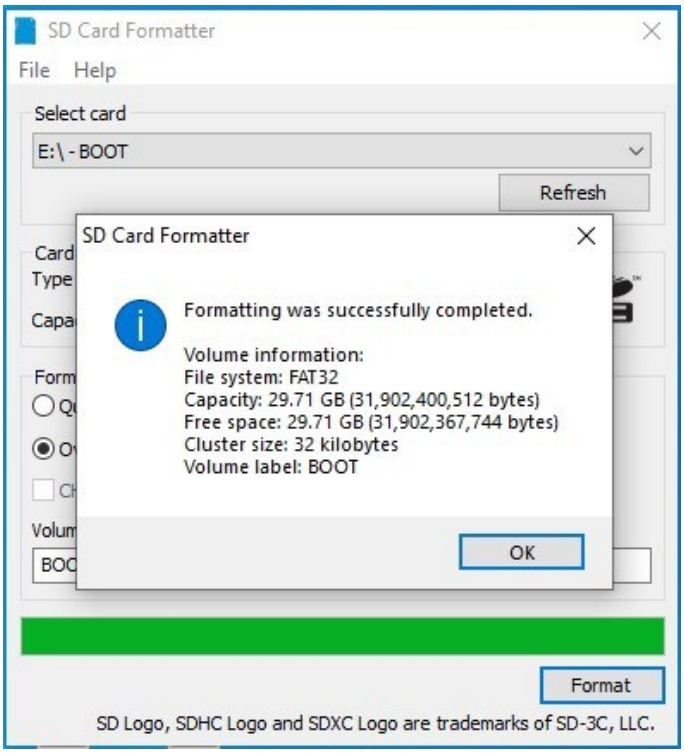

Figure 11. Completion of formatting by sd card formatter.

After formatting, the special purposed operation system must be installed into the sd card, which is noobs operational system. Installation of the system can be done with the help of special installer. There are different kinds of special purposed installers exist. For the purpose of simplicity in the thesis work as option we have used BalenaEtcher installer (figure 12).

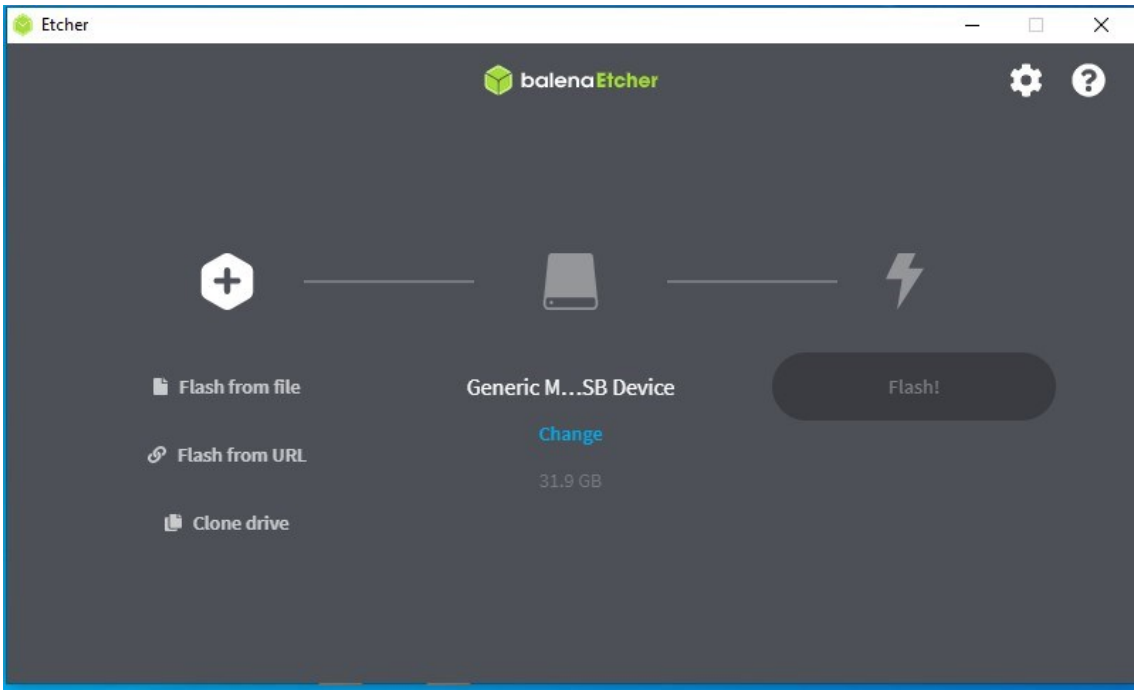

Figure 12 Special purposed installer of noobs

In order to install the noobs operational system with the help of BalenaEtcher installer it is required to download the operational system of the Raspberry Pi board. This can be done from the webpage of the Raspberry Pi, i.e., https://www.raspberrypi.org/software/operating-systems/ (figure 13).

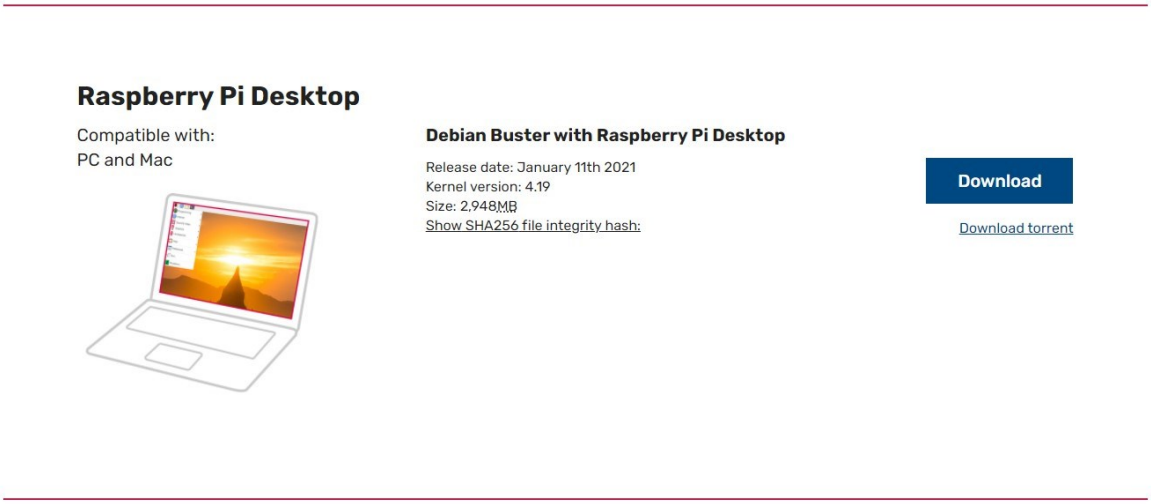

Figure 13. Website of the noobs operational system

After the downloading the noobs software, the installation process with the help of BalenaEtcher can be performed.

For the installation of the NOOBS software into the sd card with the help of BalenaEtcher, it is required to flash the file (figure 12). From the options “flash from file” must be chosen, then as a generic place sd card must be chosen. After setting up the all the parameters “Flash” button must be pressed to flash the NOOBS file into the sd card.

During the process of the flashing different kinds of pop-screens may appear and it is required not to take any action up until the end of flashing.

As the flashing finishes, in the screen “Flash complete” notes will appear. Consequently, this means that our sd card ready (figure 14).

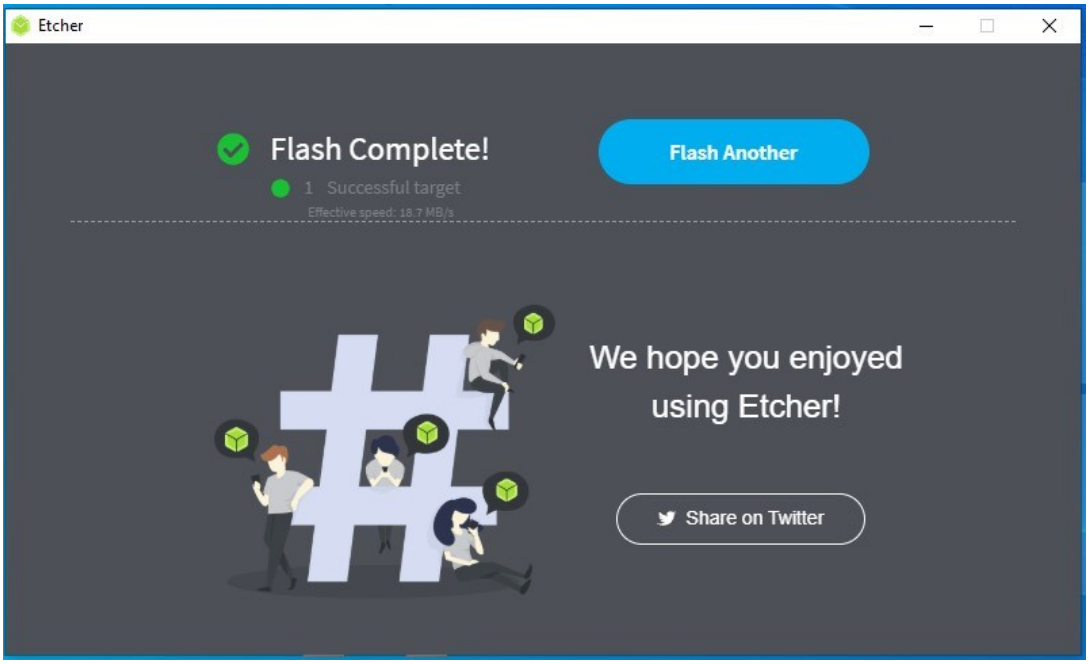

Figure 14. Illustration of complete of flashing

Special attention must be paid, after flashing finishes, computer offers as a default formatting of the sd card. One must not format the sd card, because this may destroy the files inside of the sd card.

After finishing the flashing, one has to prepare the software inside the sd card for the connection of the Raspberry Pi via internet media to any computer. This process can be done with the addition of the 2 files into the content of the sd card.

In the content of the sd card txt file must be created and written these lines of codes (figure 15).

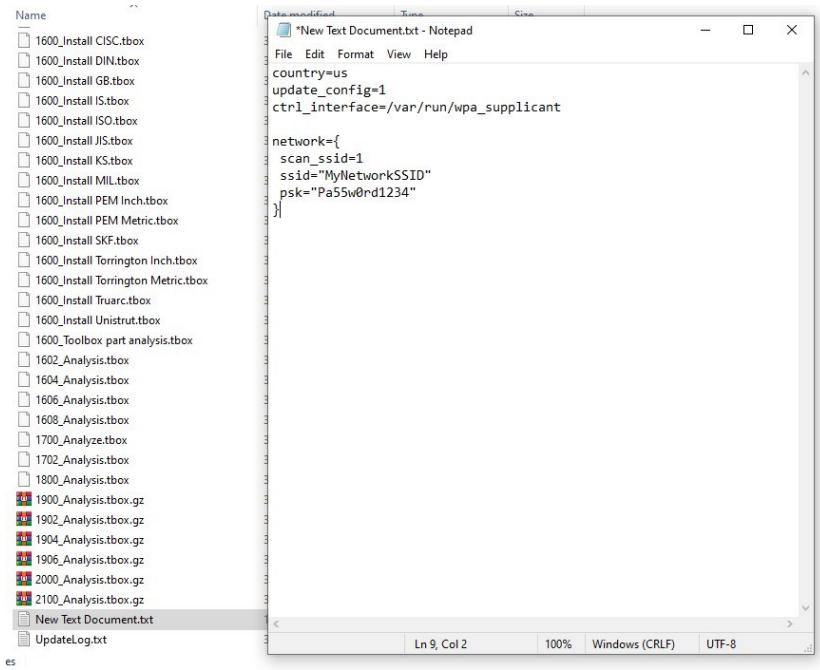

Figure 15. Creation of additional files in sd card.

The country must be changed to the one in which is required. The “ssid” is the internet network name and “psk” is the password to the internet network. The above-mentioned parameter must also be changed according to the need.

After changing the required parameters, the file must be saved without any extension under the name of wpa_supplicant.conf. The deletion of the extension of the file can be done in changing the name of the file (figure 16).

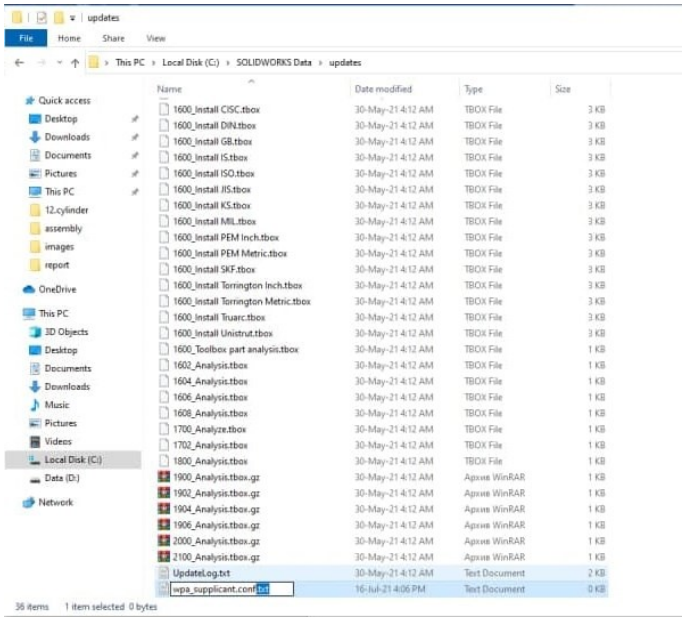

Figure 16. Changing the extension of the wpa_supplicant.conf

As a final step for the preparation of the sd card the SSH file must be added. In order to add SSH file in the content of the sd card empty txt file must be created and saved under the name of SSH without any extension. This SSH file will serve as a bridge in the connection Raspberry Pi and computer.

After performing all the steps, it is time to insert the sd card to the sd card slot of the Raspberry Pi. It is essential to give some time to the Raspberry Pi to install all the software. This time may take 3-4 minutes.

In order to check whether Raspberry Pi board is ready or not, one can use command window to ping the computer with the Raspberry Pi (figure 17).

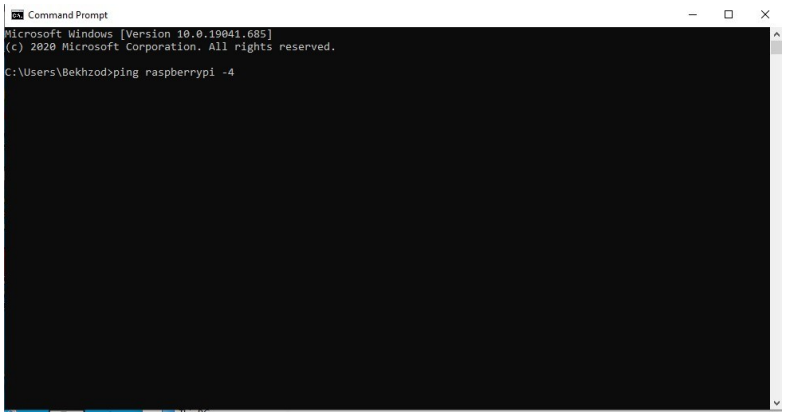

Figure 17. Pinging computer with Raspberry Pi.

If the pinging succeeds, the following message must appear in the command window (figure 18).

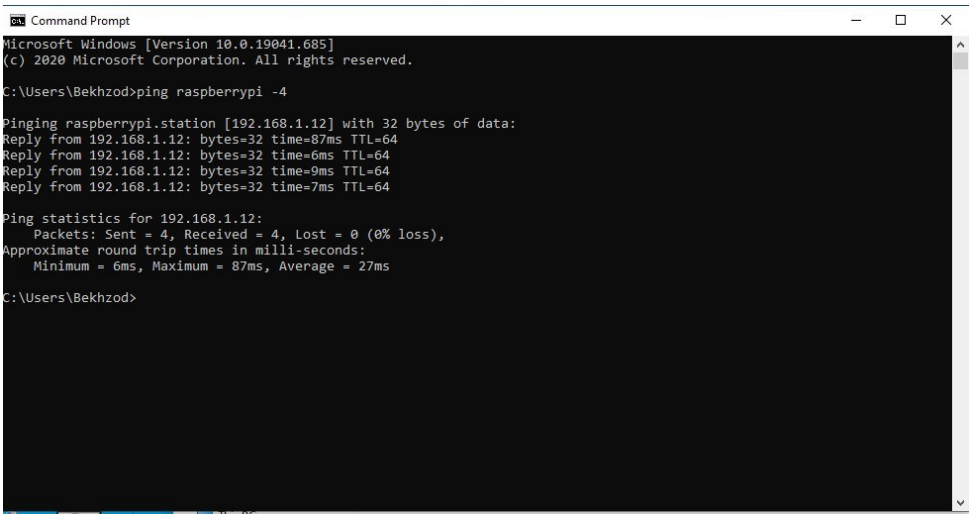

Figure 18. Pinging performed successfully.

After the successful pinging, it is time to install the library of the Raspberry Pi in the Codesys software. In order to perform this step, the library of Rasberry Pi must be downloaded from the website of Codesys, i.e., https://store.codesys.com/en/codesys-control-for-raspberry-pi-sl.html (figure 19).

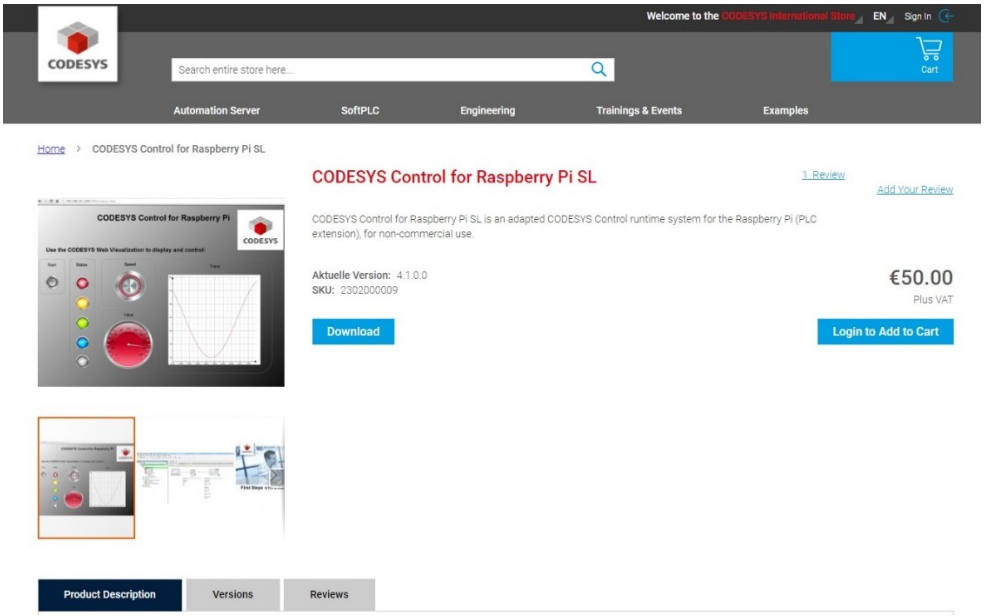

Figure 19. Web site for downloading library of Raspberry Pi

As the downloading finishes, one can start with installing the library of the Raspberry Pi in the Codesys software. To install the library, in the codesys the from the tools panel package manager option must be chosen and from the appeared windows the file downloaded as library must be chosen. The figure 20 and 21 illustrate the above-mentioned steps.

As the installation finishes, one must update the library of the Codesys software. To perform this operation from the tools panel ‘’Update Raspberry Pi “option must be chosen.

After performing all above-mentioned steps, the Codesys software is ready to make a connection with Raspberry Pi board.

As a final step to connect Codesys software to the Raspberry Pi board the parameters must be set in the Raspberry Pi block in the Codesys software (figure 22).

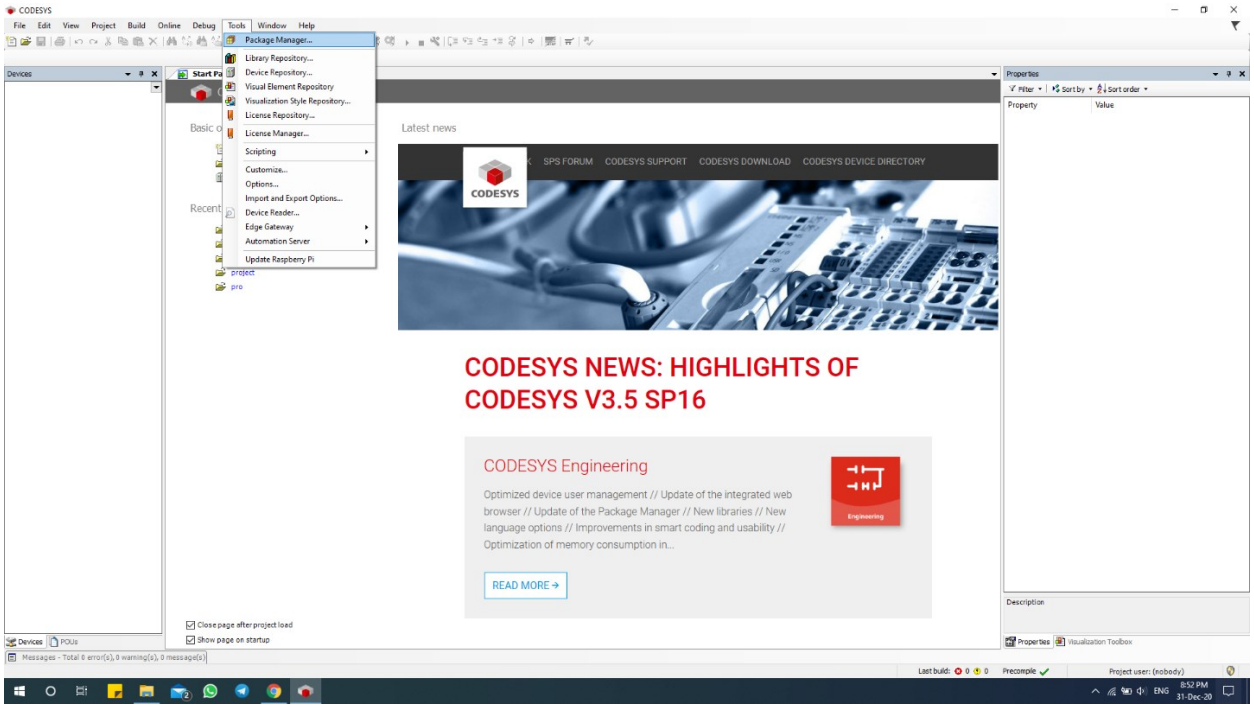

Figure 20. Installation of library

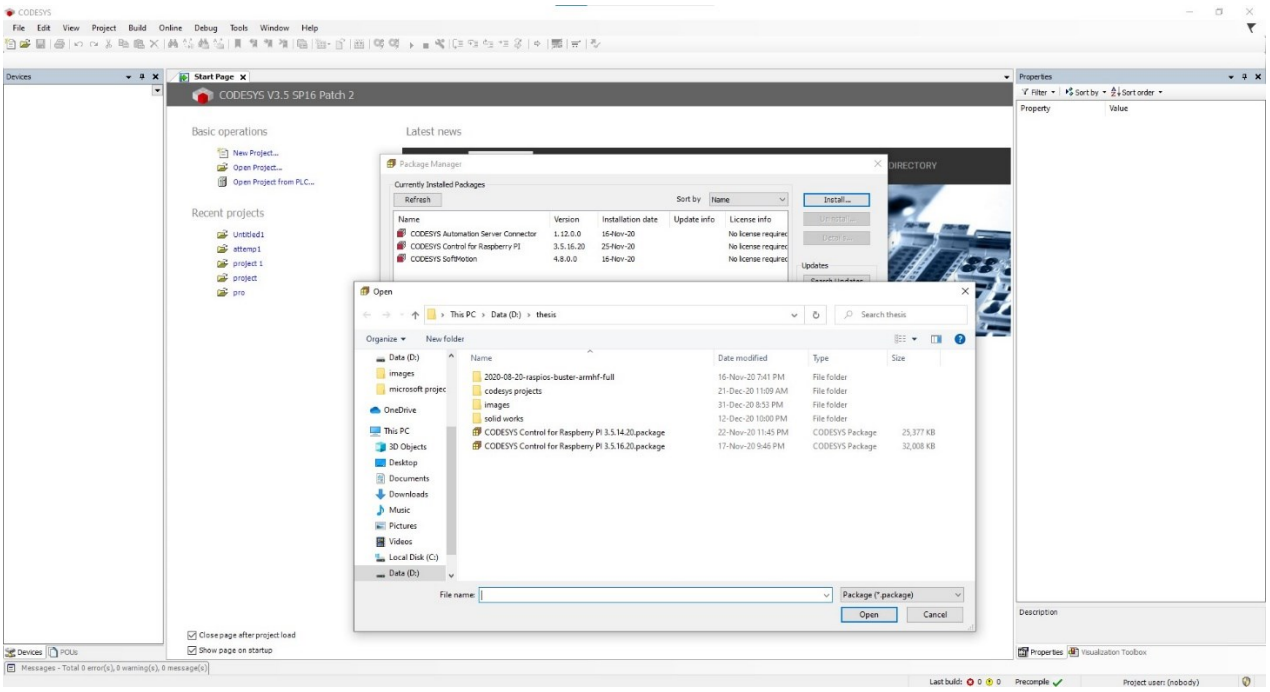

Figure 21. Installation of library.

As it can be seen from the figure 22 that one must insert in login credentials the username as “pi” and the password as “raspberry”. The login and password are default at the beginning of the connection, and they can be changes later if required.

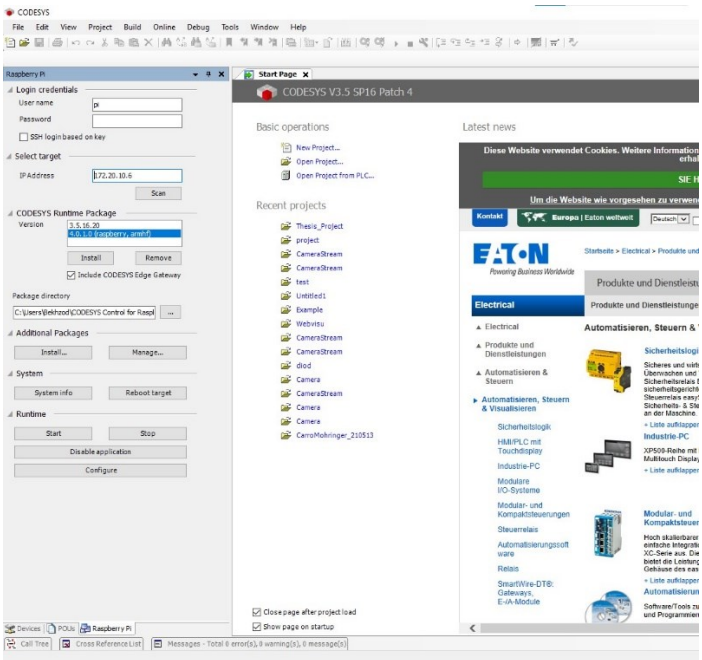

Figure 22. Setting up connection parameters.

To set the connect between Raspberry Pi and Codesys software, username, password must be inserted, the IP address of the Raspberry Pi must be scanned and as a last step the package must be installed. Special attention must be paid that the tick must be put in the “include CODESYS Edge Gateway” line, otherwise Codesys is not able to get the IP address of the Raspberry Pi.

After setting up the connection between Raspberry Pi and Codesys, it is possible to dive into programming of the Codesys software. The programming of the Codesys software can be done in various ways, like FBD, ladder diagram, SFC, ST and others.

For the programming this thesis, the ST and FBD programming style was chosen. The reason of choosing this programming style is just because of its simplicity, dexterity, and ability to control with ease the software.

3.2 IMPLEMENTATION OF COMPONENTS IN CODESYS

3.2.1 DC MOTOR

For the control of the speed rotation of the motor, the BLINK function block was used. The figure 23 illustrates the BLINK function block of the util library.

Figure 23. Blink function block

The BLINK function block consists of enable, timelow, timehigh inputs and out as output. The enable input serves as enable signal of the function block, the time low is the amount of time when the enable signal is off, i.e., zero. The timehigh is the amount of time when the enable signal is on. The out is the signal which depend on the values of the timelow and timehigh.

By varying the timelow and timehigh it is possible to produce different kinds of duty cycles, i.e., PWM signal.

As it was shown in the above hardware section that the full working voltage of the motor is 12 V. By varying this 12V full working voltage it is also possible to produce different velocities. The variation of the 12V appears by the help of the signal from the GPIO pin of the Raspberry Pi board. The production of the signal of the GPIO pin of the Raspberry Pi board is being controlled by the blink function block, that is to say, by controlling the timelow and time high it is possible to produce PWM signal with different duty cycle.

The assignation of the value for the timelow and timehigh occur with the help Codesys visualization. That is to say, the operator is able to assign any value for the timelow or timehigh from control panel. It is possible to see the illustration of the control panel from the figure 24.

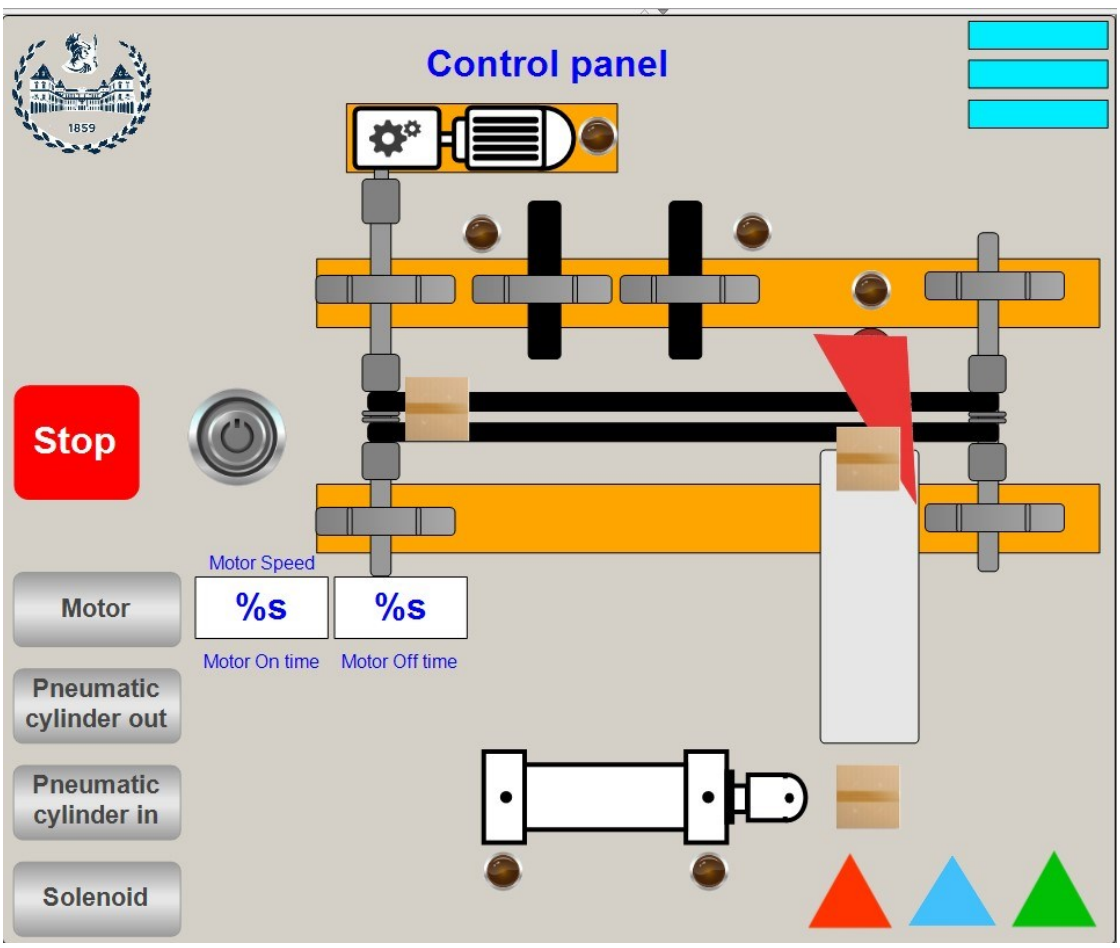

Figure 24. Control panel

The assignation of the signal to GPIO pin of the Raspberry Pi board is another quite important factor. This operation can be done from the GPIO section of the main program log. The figure 25 illustrates the place of the GPIO pin in device log.

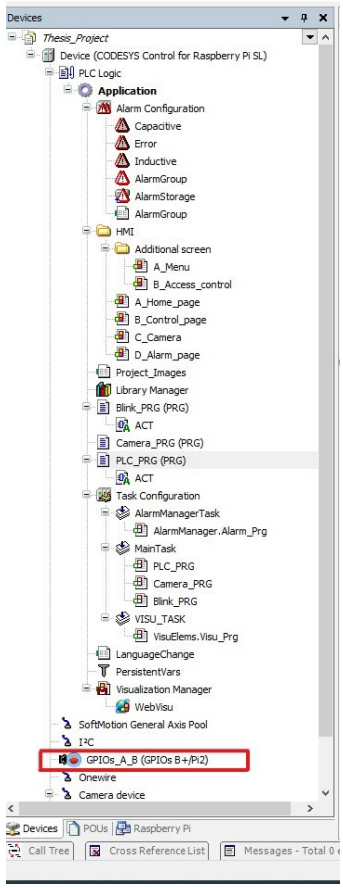

Figure 25. Device log

The GPIO pin of the Raspberry Pi must assigned as input or output. In the case of the motor the GPIO pin of the Raspberry Pi serves as a output, because it gives signal to the special designed electric schematics to control the surpass of the 12V through the collector and emitter pins of the transistor. Figure 26 illustrates the assignment of the GPIO pin for motor as output.

For the motor, pin GPIO 24 was assigned and also it is possible to see from the figure 27 that special variable is mapped to the same GPIO pin from the GPIO I/O mapping section of the GPIO of device log.

GPIOs Parameters
GPIOs I/O Mapping
GPIOs IEC Objects
Status
Information

Parameter	Type	Value	Default Value	Unit	Description
GPIO4	Enumeration of BYTE	not used	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	Input	not used		configuration of GPIO12
GPIO13	Enumeration of BYTE	not used	not used		configuration of GPIO13
GPIO16	Enumeration of BYTE	Input	not used		configuration of GPIO16
GPIO17	Enumeration of BYTE	Input	not used		configuration of GPIO17
GPIO18	Enumeration of BYTE	Output	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	not used	not used		configuration of GPIO22
GPIO23	Enumeration of BYTE	Output	not used		configuration of GPIO23
GPIO24	Enumeration of BYTE	Output	not used		configuration of GPIO24
GPIO25	Enumeration of BYTE	Output	not used		configuration of GPIO25
GPIO26	Enumeration of BYTE	not used	not used		configuration of GPIO26
GPIO27	Enumeration of BYTE	not used	not used		configuration of GPIO27

Figure 26. Assignment of GPIO pin of motor as output.

GPIOs Parameters
GPIOs I/O Mapping
GPIOs IEC Objects
Status
Information

Find | Filter Show all | Add FB for IO Channel... | Go to Instance

Variable	Mapping	Channel	Address	Type	Unit	Description
		digital inputs (GPIO0..GPIO31)	%ID0	DWORD		
		Bit4	%IX0.4	BOOL		
		Bit5	%IX0.5	BOOL		
		Bit6	%IX0.6	BOOL		
xInductiveSensr		Bit12	%IX1.4	BOOL		
		Bit13	%IX1.5	BOOL		
xCapacitiveSensr		Bit16	%IX2.0	BOOL		
xStopButton		Bit17	%IX2.1	BOOL		
		Bit18	%IX2.2	BOOL		
		Bit19	%IX2.3	BOOL		
		Bit20	%IX2.4	BOOL		
		Bit21	%IX2.5	BOOL		
		Bit22	%IX2.6	BOOL		
		Bit23	%IX2.7	BOOL		
		Bit24	%IX3.0	BOOL		
		Bit25	%IX3.1	BOOL		
		Bit26	%IX3.2	BOOL		
		Bit27	%IX3.3	BOOL		
		digital outputs (GPIO0..GPIO31)	%QD0	DWORD		
		Bit4	%QX0.4	BOOL		
		Bit5	%QX0.5	BOOL		
		Bit6	%QX0.6	BOOL		
		Bit12	%QX1.4	BOOL		
		Bit13	%QX1.5	BOOL		
		Bit16	%QX2.0	BOOL		
		Bit17	%QX2.1	BOOL		
xCylinderOut		Bit18	%QX2.2	BOOL		
		Bit19	%QX2.3	BOOL		
		Bit20	%QX2.4	BOOL		
		Bit21	%QX2.5	BOOL		
		Bit22	%QX2.6	BOOL		
xCylinderIn		Bit23	%QX2.7	BOOL		
xMotorOut		Bit24	%QX3.0	BOOL		
xPushPullSolnd		Bit25	%QX3.1	BOOL		
		Bit26	%QX3.2	BOOL		
		Bit27	%QX3.3	BOOL		

Reset Mapping | Always update variables | Enabled 1 (use bus cycle task if not used in any task)

= Create new variable | = Map to existing variable

Bus Cycle Options
Bus cycle task: Use parent bus cycle setting

Figure 27. GPIO pin for motor

3.2.2 INDUCTIVE AND CAPACITIVE SENSORS

Application of inductive and capacitive sensors in the Codesys sensor appear in a different way. The reason of being so is they function as input rather than output. For this reason, the assignation of the GPIO pins must be done as input. Figure 28 illustrates the assignation of the sensor in GPIO mapping as input.

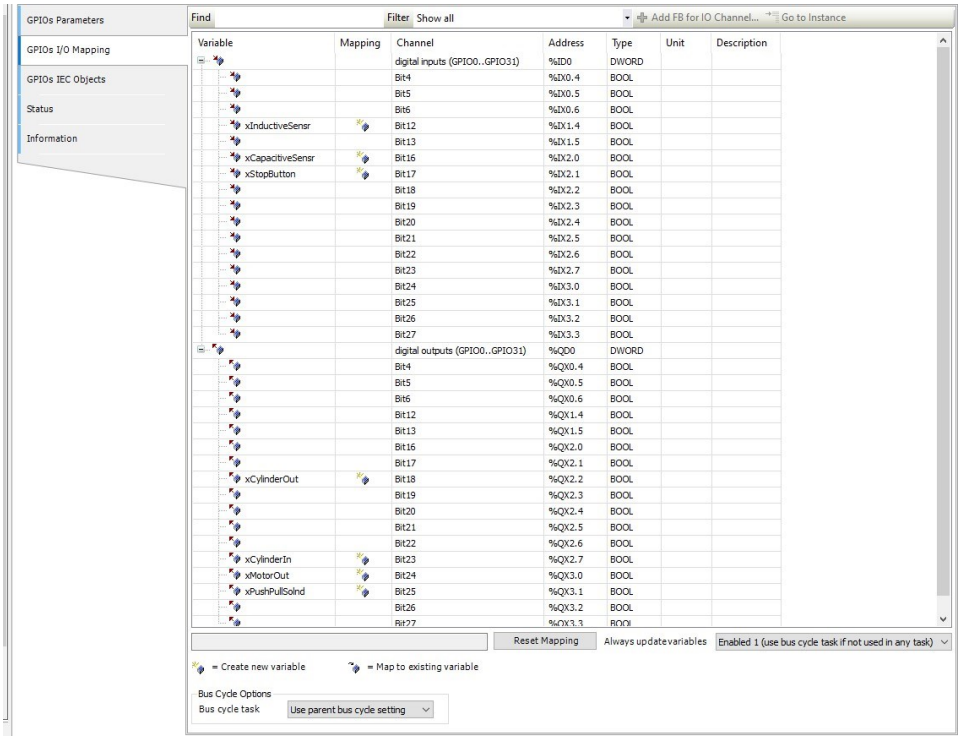

Figure 28. Assignation of inductive and capacitive sensors into GPIO

For the inductive sensor, the GPIO 12 was assigned and for the capacitive sensor the GPIO 16 was assigned.

Moreover, to ensure the functionality of the sensor 2 lamps were associated with these 2 sensors. Figure 29 and Figure 30 illustrate the association of the 2 sensor signals. These 2 lamps will blink when there is signal in the signal pin of the inductive and capacitive sensors.

By the blinking of the lamps, it is possible to determine what kind of object was placed in front of the sensors, i.e., metallic or plastic object.

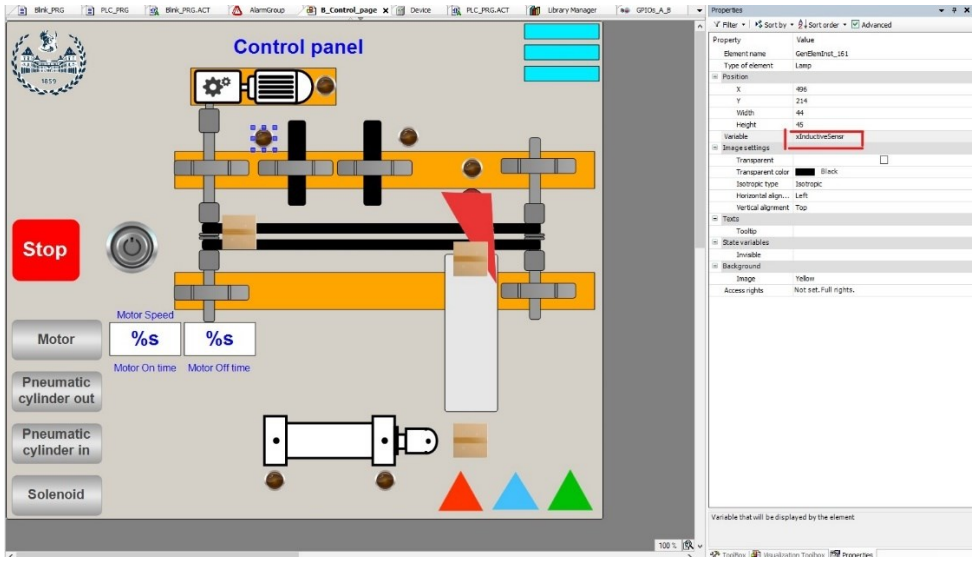

Figure 29. Assignation of the inductive sensor signal in control panel

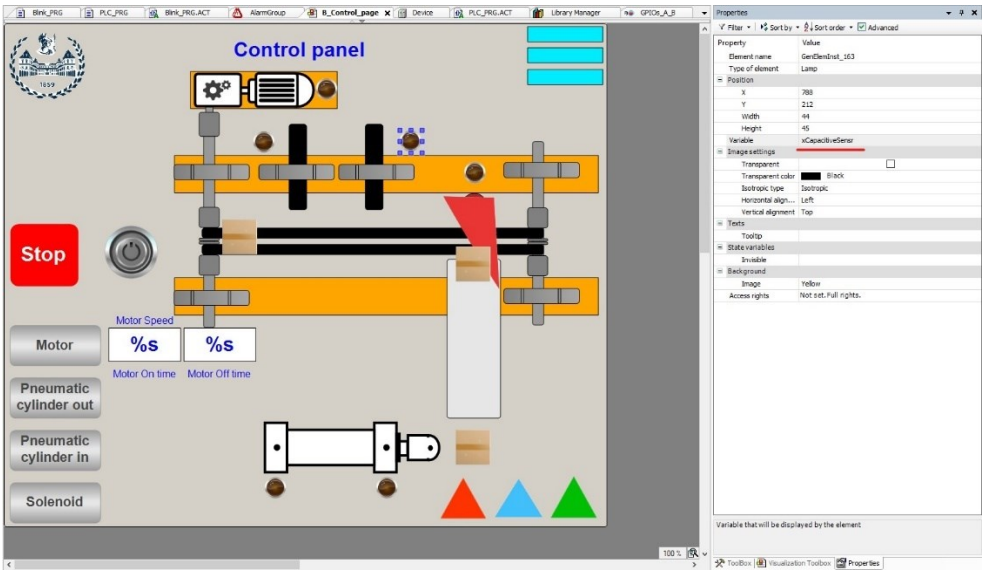

Figure 30. Assignation of the capacitive sensor signal in control panel

Moreover, it is also possible to change the blinking color of the lamps when there is signal in the signal pin of the sensors. By default, the lamps will blink as yellow, but it is possible to change them as red, green, blue and grey depending on the need.

3.2.3 PUSH-PULL SOLENOID

The control of the push-pull solenoid is performed from the GPIO 25 and special variable is assigned for this GPIO 25.

Moreover, this special variable is connect with user friendly button in the visualization of Codesys software (figure 31).

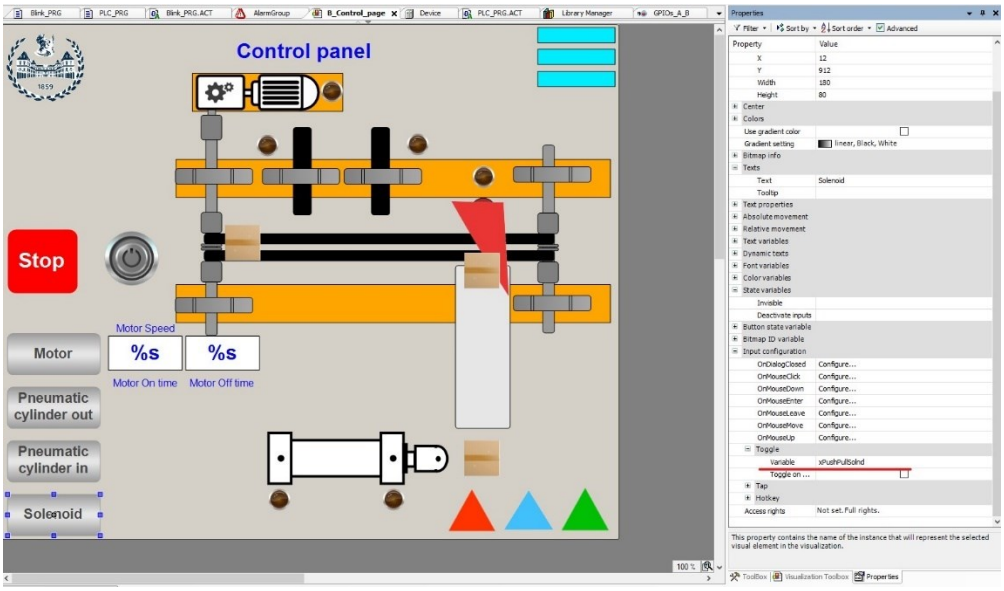

Figure 31. Assignment of solenoid control variable

The working principle of this button is as follows. If the button is pressed it will remain as “True” even if it is released. To make the button into the “False” condition it is required to press the “Solenoid” button one more time.

3.2.4 ELECTRO-PNEUMATIC CYLINDER

The control of the pneumatic cylinder is performed by the help of electro-valve. As it is well known that the electro valve works in 24V. This 24 V is control by the help of special designed electric circuit shown in the hardware section.

For the control of the 2 terminals of the electro-valve 2 pins of the Raspberry Pi must be assigned. These 2 pins are GPIO 18 and 23. The GPIO 18 serves for the extension of the pneumatic cylinder piston and GPIO 23 serves for the instroke of the pneumatic cylinder piston.

Assignment of 2 variables controlling the instroke and outstroke of the cylinder piston can be seen in visualization page of Codesys software (figure 32).

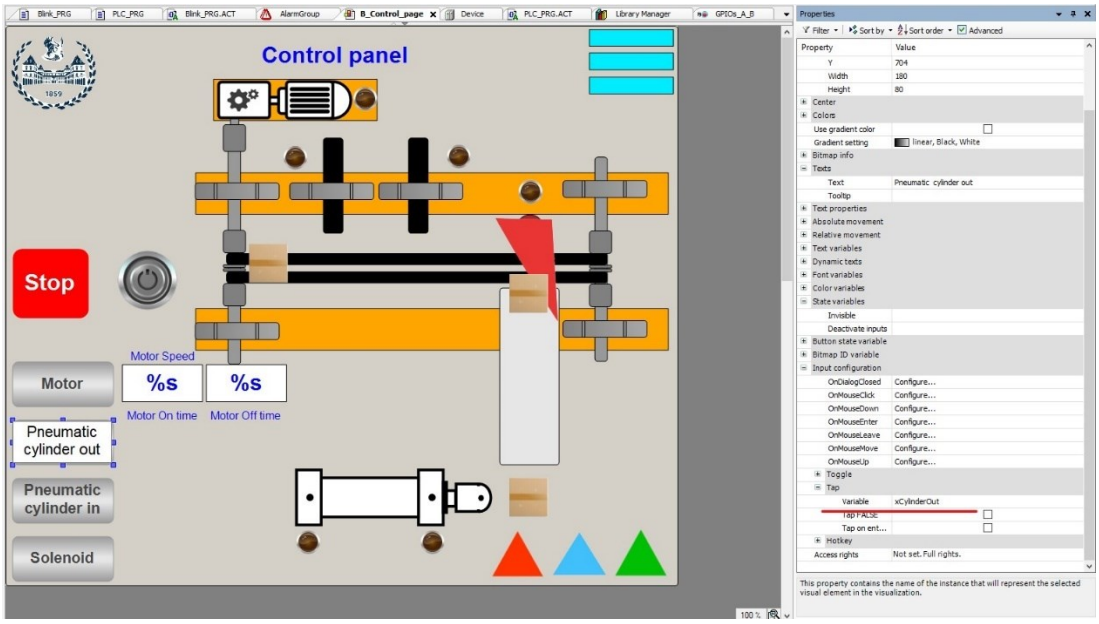

Figure 32. Assignment of the variable to control movement of cylinder piston

It is worth to notify that, specially assigned variables for the control of the cylinder piston movement must be placed in the “Tap” section. That is to say that the variable will remain true as the button is pressed and held and whenever the button is released the value of the variable will go to false state.

3.2.5 RASPBERRY PI CAMERA

The control of the Raspberry Pi camera appears through the web browser visualization object (figure 32).

Raspberry Pi camera has special IP address in which the camera works on live mode. This IP address is the IP address of the Raspberry Pi and htm, i.e., IPaddress/htm.

In order the web browser to show the on live, it must be activated. This activation can be done through the show section control variables (figure 33).

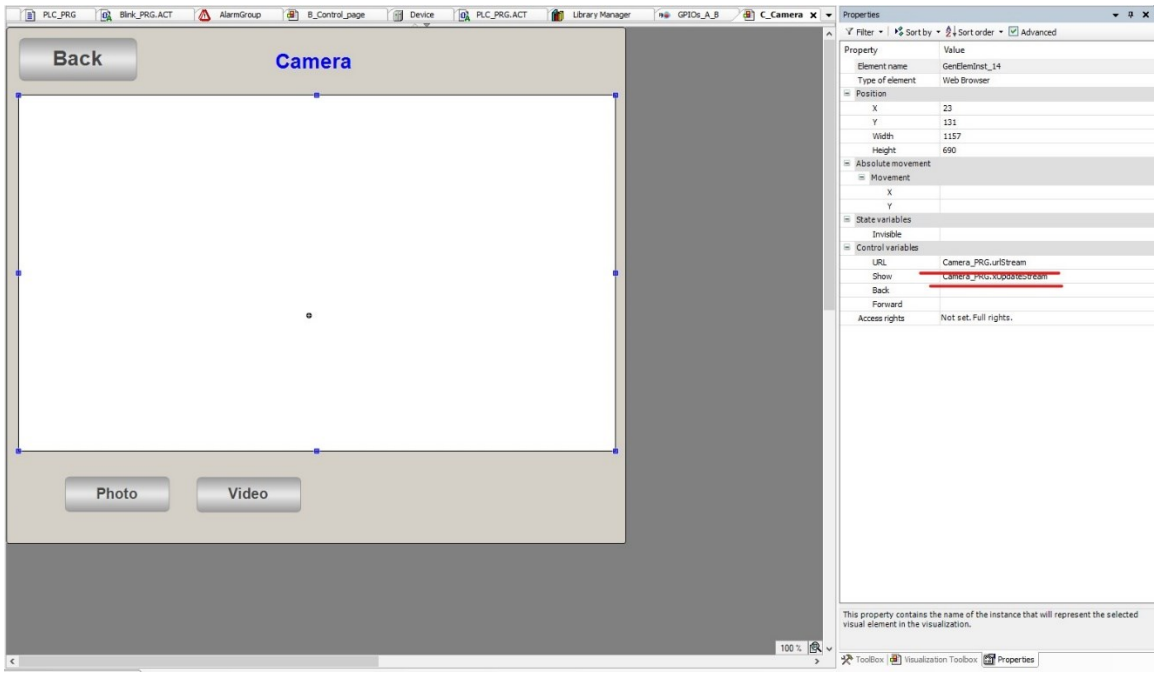

Figure 33. Web browser in the visualization

Moreover, it is possible to record and take picture whenever it is required. This operation can be done by single line of code. That is to take picture Raspberry_Pi_Camera.Still('-o Picture.jpg') and to take video Raspberry_Pi_Camera.Vid('-o Video.hd').

The two button below the web browser will serve for taking picture and taking video.

4. CONCLUSION AND FUTURE PERSPECTIVES

It can be seen from the thesis works that the potential of the Raspberry Pi boards is beyond the imagination, and it has more applications in the future.

The evolution of the Raspberry Pi boards will lead to the various application of the Raspberry Pi boards.

Nevertheless, it is also possible to connect multiple Raspberry Pi boards and use them as PLC controllers to perform multiple operations. But this is another topic of discussion.

BIBLIOGRAPHY

[1] https://www.autodesk.com/products/eagle/blog/how-an-optocoupler-works/

[2] https://www.raspberrypi.org/software/operating-systems/

[3] https://store.codesys.com/en/codesys-control-for-raspberry-pi-sl.html

[4] https://www.youtube.com/watch?v=dhY8m_Eg5iU

[5] https://www.raspberrypi.org/forums/ --- multiple discussions

[6] https://forge.codesys.com/forge/talk/ --- multiple discussions