2. Case packaging machine CP4800

2.1.Introduction

CP4800 is a case packaging machine, which is putting products into cases. Originally it is a part of the system designed for Harndrup Frugtlager A/S. CP4800 together with CS1200 palletizing robot is a complete end-of-line packaging solution. Robot and case packer have common steering system. My task was to make CP4800 standalone machine.

2.2.Machine description

CP4800 with CS1200 is a complete solution for packaging fixed size trays, punnets or bags with products (for example tomatoes or onions). Trays are coming into CP4800, where two layers of eight products are put into plastic box (600x400mm). Filled box is coming out from the machine and is palletized by the CS1200 robot. Case packer is feed with empty boxes by the same robot, which is taking them from pallet with empty boxes (depalletizing).

Fig. 1 Case Packer CP4800

CP4800 is a simple machine. Trays, after leaving flow packer, are coming on narrow conveyor very close to each other. Just before CP4800 there is short “acceleration conveyor”, which is running faster – to make spaces between products. Trays are stopping on the mechanical stop and from the pushing place are pushed inside the machine. Two trays are pushed inside at one time. Two photosensors are

mounted on the pushing place to sense ready to push products. When the trays are being pushed, another are stopped by the side of the pusher, to avoid them from entering the pushing place. About 70 products can come into machine per minute. Inside the machine trays are placed on two metal plates. After pushing 4 times, 8 products are inside. Metal frame inside is pushing trays to the sides to put them close to each other. In the next step metal plates are opening to the sides, and products are falling down to the box below. They are also pushed from the top of CP4800, to avoid them from stocking on the way down. Box is going very fast up and down to create shake effect – trays should stay exactly on the bottom of the box, not on top of each other, to be ready for the next layer. Metal plates are closed immediately after products felt down, so the pusher can put new products inside almost without brakes. After putting two layers, filled box is going out from the machine and new one is pushed inside. To better imagine the whole process please visit company's website[1].

2.3.Steering system

2.3.1. Panel PC

As standalone machine case packer had to be separated from the robot cabinet. I was obligate to use panel PC as a control system. All MPN's machine are running from panel PC's with touchscreens and they have similar user interface. It means that panel PC is at the same time HMI (Human-Machine Interface) and main control system. Choosing Panel PC I didn't have to write software from the beginning, but just make suitable changes. It also gives possibility to create friendly user interface. Bad thing about Industrial PC's it that, they doesn't have any inputs/outputs. By saying input/output I mean mainly single digital signals from sensors (like photo or temperature sensors) and to actuators (like air cylinders), not PC's communication port like Ethernet or USB port, which obviously most of the PC's have. Case packer CP4800 is steered by air – it requires lot's of inputs and outputs for air cylinders. Best choice for managing inputs/outputs is PLC. If we want to extend functionality of our PC we must buy modules usually produced by other companies. With PLC we can get all modules from one manufacturer and don't have to worry about communication or power supply for the modules. PLC's extension modules from the same product line have usually standardized sizes so it is easier to mount them in electrical cabinet.

In the first robots MPN was using PLC's for managing inputs and outputs. To communicate with PC, PLC had some inputs/outputs connected to I/O's of motion controller. Trio Motion Controller is able to communicate with PC via Ethernet. So PC was communicating with PLC not directly. This was complicated and not efficient way to

[1] http://www.mpnrobotics.com/en/20/products/cp4800.html?area=movie

manage inputs and outputs. It require to write 3 programs: for PC, motion controller and PLC. Later on MPN's choice was Telemecanique Advantys.

2.3.2. Advantys

Advantys STB[2] is a distributed I/O platform. The whole segment is called Island. Main module in the island is Network Communication Module. After NIM we can easily connect many different functional modules: digital and analog input/output modules, power distribution modules, motor control modules and other specialized modules. What is interesting about this system that we are allowed to change failed I/O modules without shutting down the machine.

Fig. 2 Advantys STB Island

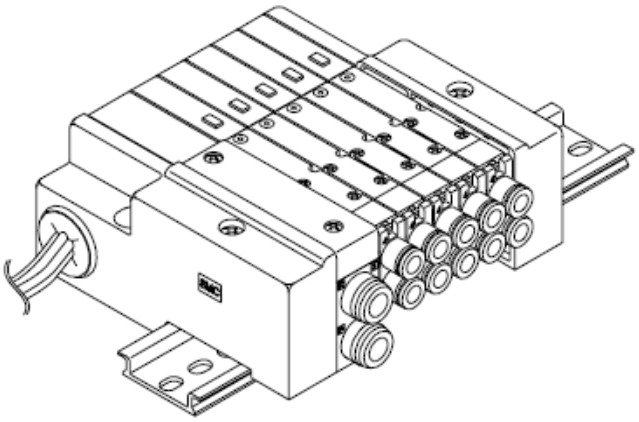

Fig. 3 SMC Manifold

I have used Advantys, because there is written driver for communication with MPN's software. I had to build my own island. Some of the modules were on stock, the rest was ordered. CP4800 requires 23 digital inputs, 16 outputs and 3 motors. My island consists of 8 modules:

- 1x NIP 212 – Ethernet communication module
- 1x PDT 3100 - power supply module for inputs and outputs
- 4x DDI 3610 – 6 digital inputs module
- 1x DDO 3705 – 16 digital outputs module
- 1x EPI 2145 – up to 4 motors can be connected

Additionally is needed mounting terminal for each module, termination for Island end, connectors for wiring input and outputs and Tesys motor drives

[2] http://www.us.telemecanique.com/products/automation/Distributed_IO/Advantys_STB/index.html

Fig. 4 Wired Advantys Island for CP4800 with connected via Ethernet cable Tesys motor drives

It is possible to use cross-over Ethernet cable to connect Advantys directly to panel PC or use straight cables with Ethernet switch between.

2.3.3. Air cylinders

CP4800 uses a lot of pneumatic cylinders as linear actuators. They are controlled by valves and manifolds manufactured by SMC which are connected to Advantys output modules. Some of air cylinders has position sensors which are sending signals to Advantsys' digital inputs. Air cylinders have only 2 possible positions – with stroke inside and outside. It is one of the reason why CP4800 is working with only one type of boxes. In CP6000 box lift is controlled by servo, so box position (height) can be perfectly adjusted. With air cylinders it is also not possible to control speed and acceleration of motion very accurate – only by regulating air pressure. But they are cheaper than electric actuators and they are good enough for this solution. Air cylinders have different lengths.

Idea for the future is to use SMC Serial Interface Units ex250 or ex500. This solution is eliminating Advantys I/O's. Those units can communicate with panel PC via Ethernet/IP open source protocol. They require also less wiring.

2.3.4. Motors

CP4800 has 3 motors: one inside the machine for conveyor transporting out filled boxes and two products transporting conveyors. All of them are 3 phases 400 VAC. They are controlled from Advantys EPI 2145 module. This module is connected to Tesys drives via Ethernet cables. Tesys is responsible for giving power to motors and protect them from trip. Tesys consists of modules which can be built like Lego. We can choose modules with different adjustable maximum output current – depending on our motor

specification. It is possible also to run motor in two directions with Tesys – but this is not required for CP4800. If motor is tripped, adequate message is showed on panel PC.

2.3.5. Photosensors

Case Packer has 3 Omron photosensors. Omron is producing wide range of photosensors with different parameters like working distance or PNP - NPN.

2.3.6. Safety and interface

During steering design we cannot forget about safety issues and interface. One of the most important buttons on every machine is emergency stop. This button should cut all power to outputs when pressed. It should stop motors and all actuators.

Another common thing for industrial machines are traffic lights. They can in a simple way show operator in which state machine is: emergency stop pressed, pause, error or running. Traffic lights should be mounted on top of the machine, that operator can easily see them from distance and react if necessary

Industrial environment is not friendly for the machine. In case packer vegetables and fruits can be smashed and damage steering system. All the devices are made for industry, so they should be resistant for those kind of situation. But even dough industrial PC is designed to work in dusty environment it should be mounted properly. Some of mounting cabinets has special exchangeable filters, which protects from dust and allow air circulation. It is important not to overheat devices inside cabinets.

We cannot forget about power switch for the machine, eliminating unplugging the power cable when it is necessary to turn off the machine.

2.4.Software

2.4.1. Linux

MPN's software is based on Linux environment. They are using Fedora Core 6 for programming. I hadn't got many experiences with Linux operating system before. During internship period I have learned basics of Linux, especially use of terminal.

Main program, which is running on MPN's machines, is written mostly in C and C++. This application is called Robostacker. It is able to control all MPN's machines: robots, case packers, checkweighers, pallet and case magazines. Robostacker is running on the machine’s panel PC's from Flash card. Flash cards are easy to exchange, which can be used for software update of the machine. Each Flash card has 2 partitions: first one with small Linux (about 7MB), second with main program(few MB).

2.4.2. Robostacker

2.4.2.1. Introduction

Main idea of this application is to control all machines. Compiled program is copied to the machine. All machine details required by the program are loaded from database. If we want to makes changes in the machine's programs, we just simply adjust database. This approach eliminates re-compiling of the Robostacker after each change.

2.4.2.2. Modbus - communication protocol

Robostacker is gathering information from external devices via Ethernet. Modbus TCP/IP[3] protocol is responsible for handling communication. It belongs to fieldbus – family of real-time industrial computer networks protocols. Modbus TCP/IP provides a reliable data transport between devices. It is open protocol based on Modbus Protocol developed by Modicon in 1979. Protocol is working on standard Ethernet infrastructure, so we are allowed to use for example standard Ethernet switches. Many industrial devices from different vendors are Modbus compatible. All devices in the network have static IP. On the other hand use of Ethernet as a communication medium, gives possibilities of distributed management. It is possible to login into machine and make changes in software from other part of the Globe – machine just need Internet connection.

It is possible to test Robostacker program on normal PC. Because it needs connection to specified hardware, it has to be run in simulation mode. In that mode all external devices like Advantys or motion controllers are simulated by software.

Machines can only be programmed off-line, but it is possible to adjust parameters on-line. Every time we need to change something in flow of the program, either we take out the flash and connect it to PC, either connecting to Panel PC via Ethernet. Remote connection is done by Linux – login to the machine – and mounting file system. Programmer can copy all required changes from his own PC.

Each device in the network requires driver. Drivers are written in C language and compiled with Robostacker. The reason that all devices need individual driver is that they need to talk with Robostacker. All devices has their individual set of messages, status, registers and errors sent using Modbus TCP/IP.

2.4.2.3. Database

During every machine boot-up Robostacker is reading important machine parameters from the set of text-files called 'database'. Files are read line by line into memory. Most important catalogues in database are:

[3] http://www.modbus.org/

- system – file system.ini contains information about machine details like: id, location, panel PC and Advantys IP address, touchscreen calibration values, access level, main interface dialog
- workcell – file *.workcell contains list of all other files from database which should be loaded during boot-up
- statemachine – files *.statemachine describe whole behavior of the machine. Each file is responsible for small piece of the robot. Company has developed their own language for writing statemachines. It has functionality like any other programming language: definition of variables and constants, logical operators, conditionals, etc. Every statemachine contains many states and can be in one state at a time. Every statemachine contains set of standard states like Halt, Idle, Reset, Timer, as well as set of individual states. During program execution statemachine is jumping between different states, depending on actual conditions. It can test on inputs like photo sensors or other statemachines. Programmer can define many statemachines. Sometimes statemachine is only responsible for controlling single sensor. Use of statemachines is very clever, cause it makes programming more similar to object-oriented. Each part of the machines is acting as independent object. I have taken all statemachines connected with case packer from old system in Harndrup. I had to write also few new statemachines to run CP4800 as independent machine.
- dio – file *.io contains information about Advantys, like island's IP address, number and names of modules. It also links all inputs and outputs with their individual names, which are used in statemachines. I had to create *.io file corresponding to Advantys Island
- language – file *.language contains languagepack for GUI. It is possible to switch languages from menu
- statetekst – file *.statetekst contains machine status messages displayed on the screen. They are also in different languages
- item – file *.item describes properties (dimensions) of box in which products are being packaged.
- cppatern – file *.cppatern contains information about number of products and layers in the box

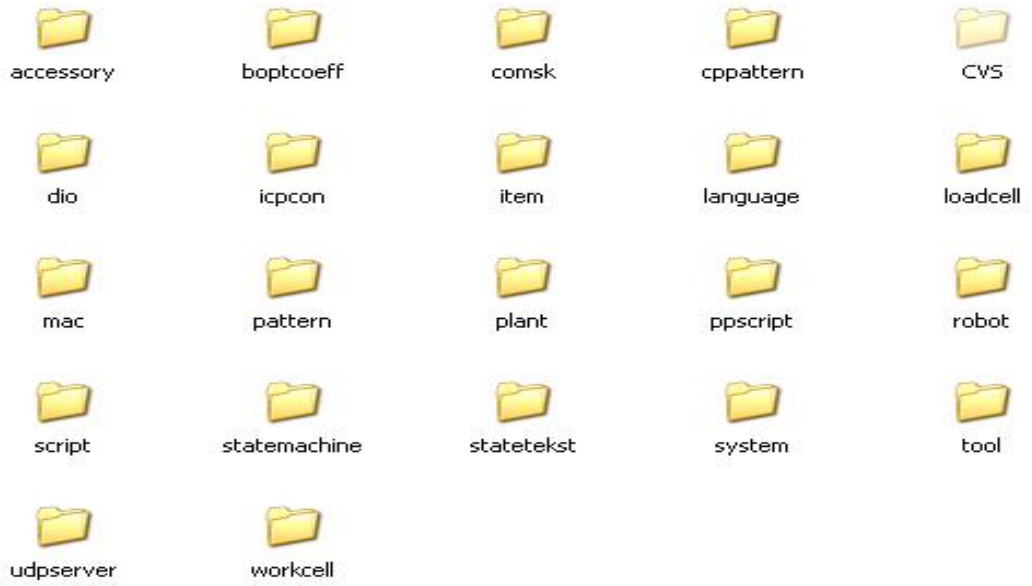

Fig. 5 Database

2.4.2.4. GUI

Big advantage of MPN's machines is color touchscreen. It gives possibility to create advanced and dynamic Graphical User Interface.

Touchscreen size is 12", so user interface shouldn't be too complicated and contain too many information at one time. Buttons should be big enough to press it with finger. Sometimes workers are wearing gloves.

There are different access levels – some options are only available for supervisors or service. It provides necessary security and possibility of machine service using user interface.

Each MPN's machine has similar user interface: colors, shape of buttons and their placement on the screen. It is easier to operate different machines in one factory for personnel. So it was very important to keep the existing GUI. There is possibility to change language, so machines are ready to be sold abroad.

Use of panel PC gives possibility to do even more advanced and user-friendly HMI. One of the idea are animations and simulations of machine behavior displayed on the screen.

2.4.3. Advantys

After building physical island it needs to be configured. With hardware there is delivered Advantys Configuration Software for Windows. Program is very simple and easy to use. You just need to build your Island by choosing elements from library and upload it. Small range of IP addresses is possible to choose using screw driver, but if we are already online we can set any address to memory. Whole Island configuration is stored in communication module.

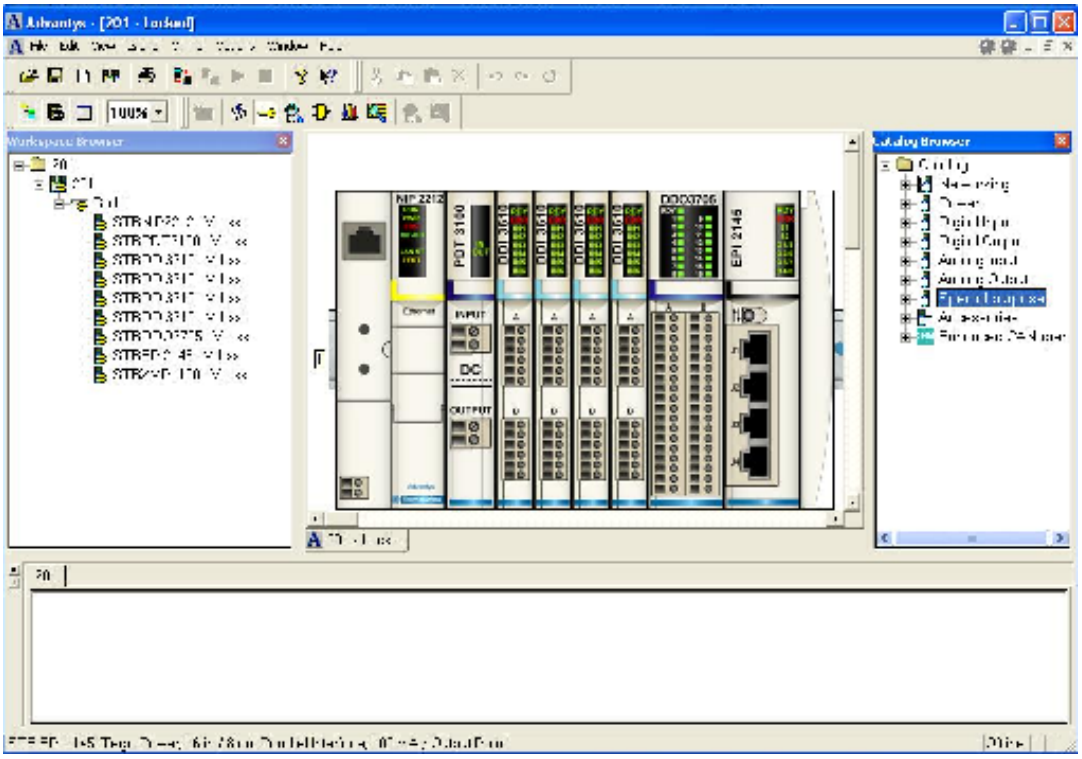

Fig. 6 Advantys Configuration Software

2.4.4. Supporting program

For easier programming MPN has developed some supporting programs. One of them is mpnguide. MPN's palletizing robots are programmed offline. Mpnguide is a program for graphical representation of programmed paths. In the program we can observe our paths and keep track of many parameters, like engines acceleration. It is very important to simulate them before testing on real robot.

My idea is to develop a program for graphical representation of statemachines. It is much easier to follow a flow of a program form a graph than from a plain text.
I made research on the Internet, because I couldn't use any of the graphs that I learned on my previous course: object oriented system analysis and design. I read a little about state diagram used to graphically represent finite state machines and found OMAC state model (fig. 8). I have also tried to find software that can translate statemachine to a graph. My choice is Graphviz. It is open-source software, already installed in many Linux distribution. There is also Windows version. Graphviz is drawing graphs from text files. It is able to draw many types of graphs and user has big control over graphs: shapes, colors, comments on arrows.

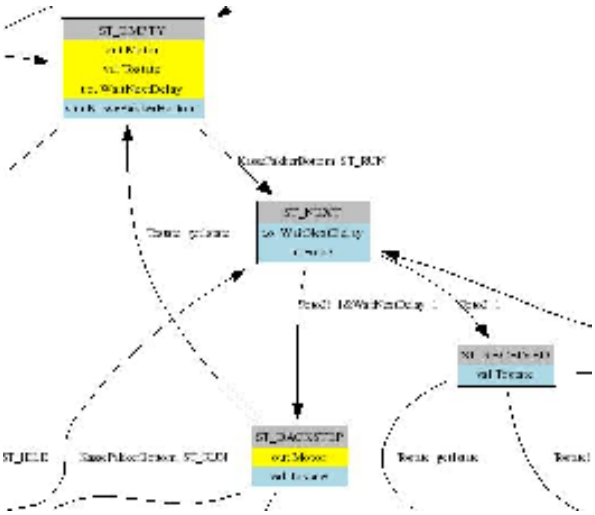

Fig. 7 Statemachine graph from Graphviz

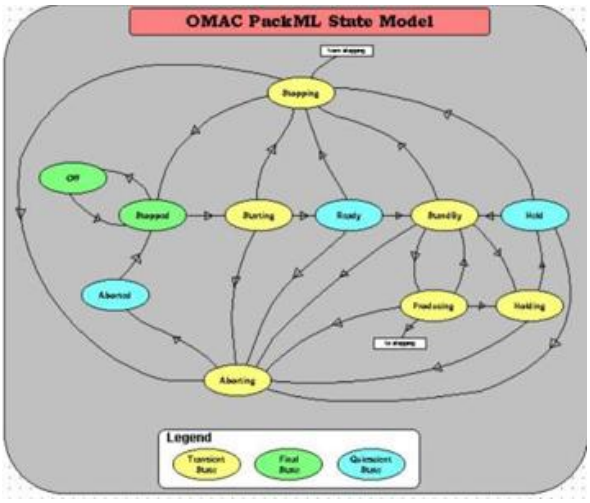

Fig. 8 OMAC state model

Idea of my program, that I called mpngraph, is to translate statemachine code into code interpretable by Graphviz. Than mpngraph should run Graphviz, which will save graph in preferable file type for printing or viewing on the screen. Graphviz supports many file types from gif and jpg to pdf. It is possible to run Graphviz from terminal.

Some code could be taken from Robostacker, cause it is already able to read each statemachine line by line. After reading each line, mpngraph should interpret it and generate suitable code for Graphviz. Mpngraph could be run from terminal or have simple GUI. Some of the statemachines are very long, so user should choose which information would he like on the graph. The most simple idea is just to show states and arrows from them to the states that are possible to jump to. Other information could be condition of state change written on arrow or inputs and outputs used in this state.

Program could be also use for machine testing, to highlight on panel PC current state.

2.5. Test

I have tested CP4800 with palletizing robot. I needed two signals between machines:

- Case packer is ready to receive empty box
- Robot is ready to receive full box for palletizing

Machines were working without any problems.

2.6.Conclusion

- CP4800 is able to work as standalone machine
- CP4800 is cheap, because it is steered by compressed air – without servo motors
- It is fast, simple and reliable
- It is not universal, because it is working with only one type of boxes