Modular control system

J. Novotný¹ and T. Beneš²¹
¹Brno University of Technology, Czech Republic
E-mail: xnovot1i@vut.cz, xbenes23@vut.cz

Abstract—This paper deals with the design, realization, and programming of a modular control system for general purposes. Intended uses in the application of micropower plant, where is needed low price. With an industrial control system it is easy to achieve a control result, but the costs are high. The embedded solution can be cheaper, but mass production and high knowledge of the system are required. Another difference is the size because the embedded solution can be like a black box, but PLC needs functionality more devices. The main goal was to design a modular control system and implement base program functionality for the next testing and development.

Keywords—modular control system, ARM, Cortex-M, CPU, industrial controller, Microchip, communication, RS485, CAN bus, micro hydro-power plant

1. INTRODUCTION

The project started with finding the best solution controlling for micro hydropower plant[2]. In the beginning, a simple control system was based on a compact solution without communication was created. This solution was perfect for this specific project, but when we tried to change the specification, there were problems. That is the reason why we have considered changing a simple compact solution to something more customizable.

The main idea of the work is to realize its own modular control system, which obtains all specific modules connected at one common internal communication bus. For example, the PLC doesn’t implement a power analyzer inside of the control system. The own solution of control system can save the place and the cost when the mass production is introduced.

This paper deals with the design, realization, and programming of a modular control system for general purposes. The project is developed in cooperation with the company ELZACO spol. s r.o..

2. REQUIREMENTS FOR CONTROL SYSTEM

The control systems usually consist of a CPU, peripherals, and a display (optional). There is generally using digital inputs and outputs in industry. Analog inputs and outputs are less represented, but they are not unforgettable. Recently, there has been a trend to avoid analog peripherals and use communications, which saves a lot of cabling and increases accuracy. This is also linked to the decentralization of the peripherals. For micro-hydro power plants the presence of the power analyzer can be an advantage.

The modular control system has an advantage, that obtains one CPU module with program and communications (optional). Peripheral modules are separated and connected by an internal bus and are independent (can be replaced by a new module without uploading the user program). The display is optional and not necessary. There are two types of display solutions:

- Separated display from the control system, which is better for decentralization, and all cables are not routed from technology to panel with display.

DOI: 10.13164/eeict.2022.126
• **All-in-one** solution is cheaper, but there is a requirement for cable routing from technology to the panel.

A separate display solution was chosen for easier organization in the switchboard of a possible combination.

Based on these parameters, the following modules were designed:

- **Processor unit** *normal* and *LITE*, which differ in the MCU and used peripherals.
- **DI module** obtains 22 galvanic isolated digital inputs configurable as *Sinking* or *Sourcing*.
- **DO module** obtains 20 galvanic isolated *Sourcing* digital outputs.
- **AIO module** obtains 8 galvanic isolated SW configurable analog input or outputs.
- **Power meter module** for measuring power network (usable for controlling in micro hydro power plant).
- **Display** with peripheral optimized for controlling small injection molding machine[3]:
  - 12 digital inputs, 12 digital outputs and 4 fast digital outputs for PWM and stepper motor control.
  - 4 analog inputs and 4 thermocouples inputs.
  - Monochromatic graphical display, USB and RS485.

All numbers of peripherals depend on the chosen case described in section 4.

### 3. COMMUNICATION PROTOCOL

The main issue is good communication between modules because the cycle of transferring all data between master and slaves modules defines the minimum cycle of the program (there is no reason to calculate with the same data). There are many *open-source* communication protocols, but they have some limitations. For the speed and the specific requirements it is better to implement own communication protocol based on available hardware. The control system consists of one master module and a few slaves modules, where data flow control by the master module is required. On the other side there are limitations by connector connecting master module with slaves modules which have only 8 (described in section 4) pins for a supply and communication. It follows from these parameters that it is best to use master-master communication, i.e. when each device has access to the bus and it also transmits on it. The access time is controlled by one master module, which allocates the transmission time to the other modules. Full duplex communication is not required as it meets the requirements and the treatment of full-duplex communication would be much more demanding.

The use of the RS485 or *CAN bus* was considered. Both of this serial communication buses are based on LVDS (Low Voltage Differential Signal) and they use only one twisted pair for transfer signal. RS485 was chosen, because it implements only physical layer of ISO/OSI model and is more flexible. CAN bus implement data link layer and not customizable, but implement more sophisticated data transferring. The own communication protocol was named as MCSBUS (*Modular Control System BUS*). Except for the common communication bus (RS485) MCSBUS implement definition of power supply (5 V) and signals for control modules (Next device program, Device program enable, Return from slave and Return to master) which is used for initialization modules (address assignment, detection of missing modules,...).

The data transferring model is similar to *open-source* communication protocol MODBUS/RTU, which is optimized for 16-bit registers. MCSBUS was resigned to work with data objects and optimization for modular system. Transmitted data have this parts:
Address for an addressing slave module. The address of the master module is 1, slave modules addresses are in the range of 2..255, and address 0 is reserved for broadcast messages from the master device (all slave devices will receive at the same time).

Control obtaining information e.g. read or write message type.

ID usable for addressing data objects.

Length of transferred object data.

Data object with required information.

CRC as data integrity check calculated from all the data above.

Inner modules communication is based on RS485, which have some limitation. In connector of MCSBUS there are still 2 pins free and they can be used for CAN bus. The next generation will use CAN bus in these free pins for faster cyclic communication, because CAN bus is deterministic and usable for events (CAN bus using multi-master bus connection and racing in the arbitration field of the message).

For example cyclic communication processor unit with digital outputs module and digital inputs module. The communication with RS485 needs to initiate start communication of every module, which means, that the master module sends a request and the slave module sends a response. The time of executing this cyclic communication is about 1 ms. With the CAN bus we can send one request message from the master module and all slave devices send a response (there is some intelligence of CAN bus because all devices send responses at the same time). This mechanism is used in CANOpen communication protocol[1].

4. DESIGN

There are various cases solutions on the market for modular solutions. The ICS variant from the manufacturer Phoenix Contact was chosen, which offers various widths, DIN rail mounting, inter-bus connectors for mounting on DIN rail, and other accessories. This system has an 8-pin internal bus connector and it offer parallel and serial connections (the connection between modules 1 and 2, 2 and 3,...). All modules are wanted to design into one size of cases (one module width) and this chosen case allows adding up to 6 connectors with 4 connections. These case connectors are designed as tool-free Push-in. Figure 2a is showing assembled PCBs of power meter with connectors. Figure 2b shows the modular system cases.

5. ELECTRONIC DESIGN

The main point of electronics is MCU, which is chosen from SAM family (Microchip) based on 32-bit ARM cortex-M CPU. For better cooperation (there are 6 different modules) it is better to chose similar MCUs (same code library). This is not possible for all modules, because some module need more performances (often declared in MIPS) than other. For modules, which need more performances ATSAME70 was chosen and as less performances MCU ATSAMC20 was chosen. The figure 1 is showing block schematic for processor unit (1a) and digital inputs (1b) modules.

6. PROGRAM

A program was written in Microchip studio with using of C programming language. FreeRTOS real-time OS (RTOS) was used for better using separate parts of functionality, which offers programmer use of tasks (with priority and round-robin scheduling), multiplexers (semaphores), queues, timers, and a lot of other functionalities. The main pros can be separate timing of inner functionality of control system tasks in contrast to user tasks (user functionality can be written or included as a separated file).

The best way to do programming of this control system is to develop own programming app with implementations of standards programming languages defined by standard EN 61131-3, but this is not easy to implement for one person developer team (better to buy PLC). Another option is to use ready-made solutions implementing EN 61131-3. One solution is OpenPLC, which runs on Raspberry Pi. Our own
control system is not needed to verify the functionality and the possibility of using it, and it is developed as a closed platform. However, in the future, this would only mean redesigning the processor unit, which is the biggest advantage of a modular control system.

7. CONCLUSION

I managed to design, revive and test our own modular control system for a general-purpose. The modular control system is functional and now is still being developed and functionality is being added. When somebody wants to use this control system, advance knowledge of programming in C language and FreeRTOS is needed, but this problem has been known from the beginning of the developing and it does not disadvantage in mass production of small hydropower plants (only one person writes and uploads the program, which is used for the lifetime of the product). In the next generations of my modular control system, shortcomings should be ironed out and additional functionality added if the control system proves to have a future.

The use of this system is not limited only as a control system, but if the control application is reprogrammed, it can be used as a remote IO over communication.
ACKNOWLEDGMENT

I would like to thank the company **ELZACO spol. s r.o.**, which offered me the background for the development and material support.

The completion of this paper was made possible by the grants:
No. FEKT-S-20-6205 - "Research in Automation, Cybernetics and Artificial Intelligence within Industry 4.0";

REFERENCES

