A DIN-RAIL BASED MODULAR DESIGN FOR CONTROLLING OBSERVATORIES AND TELESCOPES

András Pál 1,2 and László Mészáros 1

RESUMEN

En este artículo describimos un diseño electrónico multifuncional adecuado para implementar un control integrado versátil de varias funcionalidades, especialmente ideado para operar sistemas de telescopios autónomos. Nuestro sistema comprende una serie de componentes electrónicos encadenados adaptables a un riel DIN, donde cada uno de los módulos admite todas las funcionalidades relacionadas con la comunicación entre módulos, informática autónoma y conexión física, siendo los controladores compatibles con interfaces externas.

ABSTRACT

In this paper we describe a multi-functional electronics design suitable to implement versatile embedded control functionalities, especially to drive autonomous telescope systems. This system of ours is a daisy-chained series of electronics having a DIN-rail compatible form factor, where each of the modules support all of the functionalities related to inter-module communication, autonomous computing and supporting physical connection and drivers to external interfaces.

Key Words: instrumentation: detectors — methods: observational — telescopes

1. INTRODUCTION

In this paper we present the current status and development plans for an embedded telescope control system (TCS) design scheme, including a series of logic controllers, drive systems and interfaces implemented in a DIN-rail compatible form factor. The main goals of this design are both to provide an extensible structure for retrofitting telescopes as well as to implement components for new designs. Components a of TCS are a variety of embedded controllers, performing real-time functionality while communication between the various modules are realized with multi-drop bus systems. In our system, the modules support packet serialization on RS485 and CAN while media converters are designed for higher level of integration. This enables a control via Power-over-Ethernet/IP/TCP as well as wireless network extensions, being part of the Internetof-Things. The top-level protocol layer implements a memory-mapped input-output (MMIO) scheme, allowing both the controller(s) and the individual unit(s) to integrate various functionalities within the same structure of a virtual address space. This allows us, for instance, a seamless essential for the control of an autonomous telescope.

One of the drivers of our mechanical casing design is the employment and integration of the modules in conjunction with another industrial standard DIN-rail based parts such as circuit breakers, power supplies, relays or switches. We also demonstrate some use-cases for this system design, including the recently renovated, nearly hundred years old 24-inch telescope of the Konkoly Observatory. This refurbished telescope design is now suitable both for public outreach and observations related to rapid response or networking when the geographical location is relevant, such as occultation campaigns.

The structure of this paper is as follows. In § 2, we describe the main design concepts of our system. The currently available modules and the modules in the commissioning or development phases are described in § 3 and § 4, respectively. A short summary is given in § 5.

2. DESIGN CONCEPTS

To achieve a flexibility between the functionalities and the available external peripherals as well as to minimize the form factor in order to fit the full circuit in a standard (17.5 mm wide) DIN-rail mounted chassis, we split the electronics into two separate boards: a *motherboard* and an expansion board, referred as *daughterboard*. An example layout, including the "assembly process" is displayed in Fig. 1. In this example, the daughterboard is a Wi-Fi offload module, allowing the access the system bus via wireless TCP/IP connections.

¹Konkoly Observatory, Research Centre for Astronomy and Earth Sciences, Konkoly Thege Miklós út 15-17, Budapest H-1121, Hungary (apal@konkoly.hu).

 $^{^2\}mathrm{E\"otvös}$ Loránd University, Pázmány Péter sétány 1/A, Budapest H-1117, Hungary.



Fig. 1. An exhibition of the design concept, showing the various steps of the assembly of a DIN-rail based Wi-Fi–CAN bridge. The *upper left* panel shows the motherboard with the power supply, MCU, CAN interface logic and CAN physical transceiver and the terminal blocks on the power/interface side. On the right, the daughterboard is shown, contains the Wi-Fi module and some additional parts needed by the module. The *upper right panel* illustrates the assembled module, i.e. the mated motherboard-daughterboard. The *lower left* image shows the full but standalone assembly in its case having a form factor compatible with DIN-rail mounts. The *lower right* panel displays this Wi-Fi–CAN bridge along with the additional parts of an encoder subsystem, i.e. a local power supply and an SSI transceiver module.

The schematic level block diagram of the motherboard-daughterboard arrangement is shown on Fig. 2. The motherboard contains the terminal connectors for the bus communication lines, communication interface drivers, the local power supplies and a microcontroller (MCU) with AVR architecture. The daughterboard is connected to the MCU via 8 general-purpose input/output (GPIO) lines, where 4 out of this are also available as an analog-to-digital converter (ADC) input. Another two terminal connectors are used to connect 4+4 input-output lines of the daughterboard where these lines passes through the motherboard and there are not any form of physical connections. On the motherboard side, the terminal connectors of the communication lines

are identical, allowing to build daisy-chained configurations. We note here that some of the initial concepts have already been demonstrated in Mészáros & Pál (2018).

In the following, § 3, we list the available daughterboards as well as the corresponding functionalities implemented as a separate firmware running on the motherboard MCU. Current and future upgrade plans on the motherboard side can be read in § 4.

3. MODULES

In this section, we briefly summarizes our available and currently employed modules. Additional modules and variants currently being in development or test phase and another modules planned as future



Fig. 2. A schematic block diagram of the design, showing the terminal connectors (on both sides), the communication interfaces, the microcontroller and the connections between the motherboard and the daughterboard. Note that the offload CAN controller is presented only in the case of microcontrollers without an internal CAN logic interface.

upgrades are summarized in § 4. Table 1 also summarizes the available modules.

3.1. Interface bridges

While a deployed system contains mainly a single type of system bus (e.g. CAN or RS485), it is often required to have interfaces for legacy modules or modules with different types of communication protocols. For instance, such a bridge is utilized when a MEMS accelerometer-based attitude sensor is included in our system (Mészáros et al. 2014; Pál et al. 2016a), and this sensor needs a much more rigid assembly than what a DIN-rail mounted part could provide. Our interface bridges provide support for I²C, galvanically isolated RS485, CAN-RS485 and RS485-CAN bridges.

3.2. Drivers and encoders

Off-the shelf stepper motor drivers usually shipped with three logic inputs lines: drive enable, direction and step while these operates as follows. When the drive is enabled, the motor is rotated in accordance with the direction input during a rising (or falling) edge of the step input. All another parameters of the motor (such as the hold current or microstep counts) are configured usually by switches, allowing only rotation control with pre-defined steps.

Stepper drivers. Compared to interface bridges (see above), general-purpose input/output or even sensor interfacing (see below), stepper motor drivers need significantly more computing power and real-time operations are also essential at the level of microseconds. In this implementation of DIN-rail

mounted logic controllers, we followed the same control scheme here which is found in the embedded logic of the Fly's Eye Camera System (Pál et al. 2013; Mészáros et al. 2019), especially in its hexapod platform (Pál et al. 2016b).

Encoder interfaces. While the type of the physical interface can be pretty similar for incremendifferential line tal and absolute encoders (e.g. transceivers), the structure of the glue logic application running on the MCU can significantly be different. Namely, incremental encoders (such as 2-phase digital quadrature encoders or sinusoidal incremental signaling) need to operate real-time in order to avoid any miss in the steps. Contrary, absolute encoders – having, for instance, a Synchronous Serial Interface (SSI) atop an RS422 physical line – can be read out any time, in theory, without missing any step. However, SSI lines do not contain any parity or other type of information related to signal integrity. So it is recommended to poll such encoders also frequently and drop those responses which represent non-physical positions. Such an implementation is more relevant in the case of long cables and strong electromagnetic interference (e.g. in the case of dome position encoders and 3-phase dome driver motors).

3.3. General purpose control and feedback

Many applications need to provide or check quasi-static information with a size of a few bits with a relatively low bandwidth (at least, compared to higher speed communication lines). For such applications, microcontrollers can easily be programmed in GPIO mode. However, it is highly not recommended to connect MCUs directly to the physical

Daughterboard	Int. interface	Ext. interface	Galvanic	Typical	
	(motherboard)	(connectors)	isolation	application(s)	
Wi-Fi offload module	UART, GPIO	_	(by def)	Wi-Fi interface bridge	
Current sensor	ADC	$4 \times$ Hall-sensor	\checkmark	multi-channel current sensor (both for AC and DC)	
I ² C board	I^2C	Dual I^2C	_	$\rm I^2C$ -based data acquisition, sensor drivers, sensor multiplexing	
Open collector array	GPIO	$6\times$ open collectors	\checkmark	Relay driver	
RS485 interface ¹	UART, GPIO	Power, RS485	\checkmark	Interface bridge	
$RS485 \text{ interface}^1$	UART, GPIO	RS485	_	Interface bridge	
Opto-isolator array	GPIO, ADC	$4 \times$ Low-power input	\checkmark	Generic input and voltage level sensor	
Differential transceiver ²	GPIO	Differential lines	_	SSI encoders	
TTL transceiver ²	GPIO	Low-power output	_	Stepper logic	
Stepper driver	GPIO	Dual H-bridge	\checkmark	Stepper logic $+$ driver	
Synchronous ADCs	GPIO	Low-power input	_	1VPP encoder	

TABLE 1 LIST OF CURRENTLY AVAILABLE DAUGHTERBOARDS.^{*}

^{*}With their corresponding internal interfaces (to the motherboard) and external interfaces (to the terminal connectors). Note that daughterboards having the same superscript shares the same layout but some parts are not populated according to the indented purpose.

lines (due to the lack of galvanic isolation, MCU input/output pins are prone to electrostatic damage, etc.). For this purpose, we developed a few types of daughterboards, supporting simple connections to the "outside world" via galvanically isolated drivers for open-collector arrays and opto-isolators for sensing input voltage levels. In these applications, the I/O lines are simply programmed in bit-bang mode, however, there are possibilities both for broadcasting packets upon the detection of input state changes or add some basic timer-alike functionality in the case of output drivers.

These applications are widely used in critical parts of telescope control systems, such as interfacing with limit switches and controlling relays. For highly critical applications (e.g. closing a dome slit door in the case of bad weather), keep-alive, watchdog and window watchdogs (WWDGs) systems are also available: for instance a relay output is driven active only if appropriate packets are received within a certain periodicity (i.e. not too frequent but well within a pre-defined time interval).

3.4. Sensors and transducers

As we mentioned above, the available interface bridges include I^2C boards. Many types of environment sensors (temperature, pressure, humidity, thermal infrared, etc.) are shipped with I^2C interfaces where data acquisition also imply real-time sensorspecific steps. For some of these applications, the MCU firmware can also be extended with functionalities aiding data acquisition: data multiplexing, preprocessing, data accumulation with averaging, and temporary storage.

4. CURRENT DEVELOPMENTS

While the above described devices are used as a robust systems in many telescopes within the Hungarian astronomical facilities (see Table 2), we still extend the capabilities of our system in order to allow more flexible combinations for not just telescopes but any related similar automation projects. In the following, we detail the currently ongoing developments, considering the motherboard designs, additional daughterboards and alternatives for communication interfaces.

TABLE 2	
---------	--

UTILIZATION OF THE PRESENTED TCS COMPONENTS AND RELATED ELECTRONICS IN THE
1-M CLASS (AND SMALLER) TELESCOPES IN HUNGARY. $*$

Telescope	Dome	Mount	Focusing	Payload/	Redundancy
(diameter)	(+auxiliary)	(drives+encoders)		instruments	(+encoders)
Ritchey-CrCoudé $(1 \mathrm{m})$	$\checkmark\checkmark$		\checkmark		
Schmidt $(0.9 \mathrm{m})$	$\checkmark\checkmark$	$\checkmark\checkmark$	$\checkmark\checkmark$		$\checkmark\checkmark$
$AZ800 (0.8 \mathrm{m})$	$\checkmark\checkmark$				
Svábhegy Telescope $(0.6-m)$	$\checkmark\checkmark$	$\checkmark\checkmark$	(manual)	(visual)	$\checkmark\checkmark$
Ritchey-Cr. $(0.4-m)$	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark

 * Two checkmarks shows the currently operating telescope systems while a single checkmark shows the cases under deployment.

4.1. Embedded processing units

Our current series of motherboards supports AVR MCUs but the design is under upgrade in order to support the ARM Cortex M7 STM32F4xx family of controllers. Such an upgrade will extend the core sources (instructions per second, available program memory and available static RAM) by an order of magnitude. The modification imply no changes in the form factor due to the many similarities in the interfaces within these controllers.

We have demonstrated that a fully AVR2 compatible soft core can be implemented in the iCE40HX8K field-programmable gate array (FPGA) series by utilizing the $\sim 22 - 25\%$ percent of the total logic cells for the core and a few hundreds cells for peripherals like SPI or UART. We also ported successfully the code of TCP/IP-RS485 bridge (with a TCP/IP offload engine based on Wiznet W5500) as well as the stepper motor driver to an AVR2 soft core. Both ports have validated the maturity of fully FPGA-based motherboard.

However, a major drawback of an FPGA is the need of supervision circuit to enable a risk-free alternative for remote reprogramming. Otherwise, if the FPGA is enable to re-program its own bitstream configuration memory, any failure (including interrupted upload or even a tiny bug in the HDL design) could permanently brick the device, enabling only local troubleshooting. On the other hand, FPGAs provide alternatives for core logic while enabling the extension of the soft processor with custom peripherals upon request, depending on the targeted applications. Specifically, one of the key advantages of employing an FPGA is to create additional digital interfaces aiding the processing of specific inputs that might require higher speeds for reliable real-time operations. Such applications include the higher speed for 2-phase digital quadrature encoders, simultaneous recording of multi-channel analog-to-digital converters or the implementation of more robust incremental algorithms for stepper motors.

4.2. Linux support

While Linux-based systems are not targeted for real-time operations, some application processors have small requirements (considering device setup complexity, power consumption and PCB land area at the same time). Therefore, we investigated the possibilities of integrate such an application processor into our system design. Contrary to the aforementioned expansions towards ARM and FPGA soft core designs, an initial concept was planned in the form of a daughterboard instead of upgrading the motherboard. For this design, a Freescale i.MX233 processor was included in a daughterboard along with 64MB of SDRAM and a microSD card. In this setup, a Linux system running on this processor behaves as a slave offload engine, providing additional (but not real-time) extensions to the motherboard core.

4.3. Remote firmware upgrade

Despite the different architectures, both AVR and ARM cores support partial remote upgrade of their own firmware. In the case of AVR – implementing a Harvard architecture – dedicated instructions (such as LPM and SPM) are available to reach the program memory for the applications themselves and hardware-level support also assist to separate and protect dedicated areas in the program memory. These areas are called *boot loader sections* and can store code which modifies another program memory areas, hence allowing the firmware to do partial upgrades. On the other hand, ARM stores its code in the same memory area, allowing more possibilities but therefore more risks in such an upgrade. The next generation of our motherboard design contains a supervisor MCU, providing an independent in-memory system programming (ISP) interface for the main microcontroller. This allows a full upgrade and also a mutual "oversight" of the two cores, allowing a seamless management and can also be implemented in the same way as FPGA bitstream flash memory chips are programmed.

4.4. Power-over-Ethernet

While the CAN and/or RS485 interfaces are frequently implemented with auxiliary power supplies using additional wires, TCP/IP connections need additional power sources unless the both the downlink interfaces (switches) and the connected device(s) supports a common standard of Power-over-Ethernet (PoE). Although devices called *PoE splitters* exist on the market, the employment of such devices are usually cumbersome and complicates the design nearly the same way as independent power supplies would do otherwise.

5. SUMMARY

This paper summarized the key concepts of a DIN-rail form factor series of logic controllers, drivers, and interface bridges attaining the building blocks of telescope control systems. These units are extensively incorporated within the largest telescopes of Hungary, and hence, these are suitable to provide low-level control of meter-class telescopes. As we discussed earlier, we continue to extend the capabilities of these modules, providing more powerful and more generic solutions for the core logic and also expand the system with another types of connections and peripherals.

Acknowledgements: Our initiative is supported by the GINOP-2.3.2-15-2016-00033 project which is funded by the Hungarian National Research, Development and Innovation Fund together with the European Union. Partial support is received via the grant KEP-7/2018 of the Hungarian Academy of Sciences while earlier stages of the development have been supported by the grant LP2012-31 of the Hungarian Academy of Sciences.

REFERENCES

- Mészáros, L.; Jaskó, A.; Pál, A. & Csépány, G. 2014, PASP, 126, 769
- Mészáros, L. & Pál, A. 2018, Proceedings of the SPIE, Volume 10700, id. 107005M
- Mészáros, L.; Pál, A.; Csépány, G.; et al. 2019, RMxAC, 51, 116
- Pál, A., Mészáros, L.; Csépány, G. et al. 2013, AN, 334, 932
- Pál, A., Vida, K., Mészáros, L. & Mező, Gy. 2016a, Exp. Astron., 41, 1.
- Pál, A.; Mászáros, L.; Jaskó, A. et al. 2016, PASP, 128, pp. 045002