TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Alexander Raldugin 163089IVEM

HOST CONTROLLER DEVELOPMENT FOR AN OIL SPILL SENSOR

Master's thesis

Supervisor: Olev Märtens
PhD
Co-supervisors: Martin Jaanus
PhD
Viktor Alekseev
Engineer

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Alexander Raldugin 163089IVEM

JUHTKONTROLLERI ARENDUS NAFTAREOSTUSE SENSORI JAOKS

Magistritöö

Juhendaja: Olev Märtens
Doktorikraad
Kaasjuhendajad: Martin Jaanus
Doktorikraad
Viktor Alekseev
Insener

Tallinn 2018

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Alexander Raldugin

1.05.2018

Abstract

The objective of this thesis is to design a host controller for an LDI ROW oil spill sensor, based on the RCM6700 microcontroller.

This host controller should be able to communicate with the sensor either by wired PLC or wireless link, analyze data and indicate an alarm, as well as communicate with the PC using a number of industrially-accepted interfaces.

The host controller should be of industrially-suitable design, capable of withstanding various environments. The components will have to be able to provide protection from natural phenomena, working conditions, and to certain extent, human factor. The host controller should be easy to maintain and accessible for repairs or replacement, which means it should be placed at certain distance away from the sensor, while keeping a reliable link.

This thesis is written in English and is 90 pages long, including 5 chapters, 36 figures and 9 tables.

Annotatsioon

Juhtkontrolleri arendus naftareostuse sensori jaoks

Selle lõputöö eesmärk on arendada juhtkontroller LDI ROW naftareostuse sensori jaoks. Juhtkontroller ehitatakse RCM6700 mikrokontrolleri alusel.

Juhtkontroller peab olema võimeline suhtlema naftareostuse sensoriga kahel viisil, kas juhtmega liinil kasutades elektriliini-sidetehnoloogiat, või traadita raadiosidet. Elektriliinisidet tuleb kasutada väiksema distantsi korral, kuid raadioside on ette mõeldud suuremaks vahemaaks.

Juhtkontrolleri eesmärk on analüüsida sensorist saabuvaid andmeid ning teha otsus, kas on tegu naftareostusega. Seejärel peab juhtkontroller näitama häiret kasutades 4/20 mA vooluringi ning kaheseisundilist releed.

Juhtkontroller peab suhtlema arvutiga kasutades mitut tootmises ning automaatikas levinud liidest: RS-232, RS-485, USB ning Ethernet. Juhtkontroller peab tagama piisava suhtluskiiruse, et oleks võimalik kas arvutile sensorilt andmeid edastada või sensorile ning juhtkontrollerile uut tarkvara üles laadida.

Juhtkontroller on ette nähtud kasutuseks tööstuslikes tingimustes ning peab taluma erinevaid keskkondi. Komponendid peavad tagama piisavat töökindlust looduslike mõjude (näiteks, välgulöök sideliini lähedal) ning inimlikke vigade vastu. Eraldiseisva juhtkontrolleri eesmärk on tagada lihtsutatud hooldust, seetõttu peab ta võimaldama paindliku paigaldust, sealjuures sensorist eemal, kuid samal ajal peab tagama sensoriga kindlat sidet.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 90 leheküljel, 5 peatükki, 36 joonist, 9 tabelit.

List of abbreviations and terms

LDI	Laser Diagnostics Instruments
TTU	Tallinna Tehnikaülikool, *Tallinn University of Technology*
DMA	Direct Memory Access
I/O	Input/Output
GPIO	General Purpose I/O, *usually describing pins on a microcontroller*
I^2C, I2C	Inter-Integrated Circuit, *a serial bus type*
SPI	Serial Peripheral Interface bus, *a serial bus type*
MOSI	Master In – Slave Out, *SPI signal from controller (master) to an SPI device (slave)*
MISO	Master Out – Slave In, *SPI signal from an SPI device (slave) to controller (master)*
UART	Universal Asynchronous Receiver-Transmitter
PWM	Pulse Width Modulation
ADC	Analog to Digital Converter
RAM	Random-Access Memory, *a volatile memory used for storing program data*
EPROM	Erasable Programmable Read-Only Memory
PCB	Printed Circuit Board
SOC	System-On-Chip
PC	Personal Computer
LED	Light-Emitting Diode
OS	Operating System
SSD	Solid-State Drive
USB	Universal Service Bus
SD-card	Secure Digital card, *a flash memory card standard*
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
CPU	Central Processing Unit
MCU	Microcontroller Unit

IDE	Integrated Development Environment
PLC	Power Line Communication, *data transmission over the power wire*
Tx, Rx	Transmitter, Receiver. *TxD – Transmitter port D, etc.*
GND	Ground (“0V” reference point)
ISM	Industrial-Scientific-Medical, *radio bandwidth range*
RSSI	Received Signal Strength Indication
IDE	Integrated Development Environment, *software package for writing code, usually includes a compiler, debugger and other design tools*
CMR	Common-Mode Rejection Ratio, *ability of a device to reject simultaneous increase of voltage level at both inputs*
ESD	Electrostatic Discharge
EMI	Electromagnetic Interference
IC	Integrated Circuit
DIN	*Deutsches Institut für Normung,* German Institute for Standardization

Table of Contents

List of Figures

List of Tables

1 Introduction

Laser Diagnostics Instruments company, which is based in Tallinn, Estonia, has been since 1991 working on various products based on spectrography and fluorescence. One of the products is Remote Oil Watcher, which is a non-contact oil spill detector. The new model of ROW is going to consist of two parts: sensing part, which is responsible for gathering data, and analytical/communications part (host controller), that is intended to perform both data analysis, decision-making and communications. This thesis is dedicated to development of a host controller for the new model of ROW. The host controller will be built utilising Rabbit microcontroller.

1.1 State of the art

Since oil spill detectors are a niche merchandise, and the market for these devices is limited, the number of competing solutions is relatively low. Due to most of the technological solutions being classified as protected data by manufacturers, and not made available to public, only the basic aspects can be compared, such as principles of operation, interfaces and general layout.

The current ROW model produced by LDI is a one-channel detector operating on a principle of fluorescence of oil in UV radiation. The device consists of a single block, which houses both optics, electronics and communications. The interface is a 12-wire cable, which includes power, RS-485 bus connection, 4/20mA current loop interface and a relay to indicate the presence of an oil spill by closing or breaking the circuit. Data gathering, analysis and decision-making are all conducted by a single ARM processor. The device is placed in a gas-tight body.

While the current design has its advantages, there are several drawbacks to it. While all the component are in the same pressurized vessel, any modification to interfaces requires the re-design of the case. Besides, a single microcontroller performing all the operations leaves no room for additional functionality.

The new design proposes that the sensor itself would only be responsible for data gathering, accumulation and transmission to a host controller via a limited number of interfaces.

A study of competing products reveals that most of them are utilising the principle of separation of sensor and controller parts. Examples would include Chemtronic FLU-103,[1] a device that operates on the similar principles as LDI ROW, but being intended to work in closed environments such as by-pass pipes and channels, lacks flexibility and adaptivity. Another example might be ODL-1610A [2] made by Japanese company TOA-DKK. This device uses a visible spectrum laser, measuring the reflection of light from an oil film, comparing it to reflection of a clean water. A device called CRAB, made by Lumex Instruments, is an IR sensor, which has higher range, but is more affected by environmental conditions.

All of these devices have a separate controller and data gathering parts. The controller part involves a CPU, some indicators, such as LEDs or an LCD display, and interfaces to be merged into larger industrial control systems. The specific protocols and implementations are at manufacturer's discretion and are not known from open sources.

Aside from sensor implementation and detection algorithms, the major disadvantages of most competition is lower flexibility. Usually the host controller part should be installed near the sensor due to link constraints. In the proposed host controller, range should be extended significantly using a radio link. This allows for more convenient installation, with the sensor part becoming more autonomous, and additional flexibility is achieved by using a wider selection of interfaces.

1.2 Problem statement

The goal of this master thesis is to develop a host controller for a new model of oil spill sensor produced by Laser Diagnostic Instruments AS. The device is supposed to be working on a Rabbit 6000 series microcontroller. The thesis will include both hardware and software development.

The host controller will be responsible for accepting raw data from an oil spill sensor, utilizing either wireless or wired link. The data will be then processed, with the host-

controller making decision whether an oil spill is present or not. Host controller will also be responsible for transmission of this data to either PC or industrial automation equipment, supporting various communication protocols and interfaces (RS-232/485, 4/20mA current loop, Ethernet, USB, etc).

The device should be able to operate reliably in industrial environment, as well as demonstrate necessary robustness against natural phenomena or equipment malfunction. This means the host controller should have galvanic isolation at interfaces, as well as overvoltage and surge protection.

The device should be flexible enough to be integrated into working environments with various demands and conditions, which would require wide range of input voltages, low EMI and ability to withstand such factors as temperature gradient and moisture.

The development is based on current LDI AS products, specifically ROW oil spill sensor.

1.2.1 Technical requirements

The following requirements should be observed:

- Operating voltage: 9..36V
- Power requirements: Up to 10W
- Bitrate from sensor to host and from host to PC: at least 57600bps
- Supported interfaces: RS-232, RS-485, USB, Ethernet
- Alarm indication: 4/20mA loop, two-state relays
- Connection to sensor: PLC (for shorter distances) or radio link
- Additional hardware: Micro-SD card
- Support of up to 2 sensors per host
- Radio link range: ca. 300m

2 Oil Pollution Sensor Basics

An oil sensor that is being developed is a direct successor of a ROW sensor that is currently in production by LDI AS.

2.1 Theory

The sensor works on a principle of fluorescence of an oil film under UV light.

2.1.1 Fluorescence [3]

The occurrence of fluorescence is a result of a three-stage process in certain molecules and atoms (which are generally called fluorophores).

When a molecule or an atom is exposed to an electromagnetic radiation of certain frequencies, it absorbs a quantum of radiation, thus moving to a higher energy state (S_1'). This process is called excitation.

For a limited time the atom or the molecule exists in an excited state. Usually this time is within range of 1 to 10 nanoseconds. During this time, some of the energy is dissipated (the process is known as non-radiant relaxation), leading to a slightly lower energy state (S_1).

Since a molecule or an atom at high energy states is not stable, after a short time it should return to a low energy state (S_0). It can do this by emitting a quantum of electromagnetic radiation (photon). Due to energy dissipation in the excited state, the energy level is lower in the end of lifetime than initially. Therefore a quantum of radiation released will have lower energy, and since the energy of a photon is determined by its wavelength, the emission wavelength will be lower than excitation wavelength.

Since different substances have different excitation and emission peaks, it is possible to either create an emission map, which would correspond to presence of various

fluorophores in a compound (fluorescence spectrography) or to fine-tune the optics to detect only emission light with a certain frequency, the approach which is suitable for an oil sensor (or any fixed compound detector).

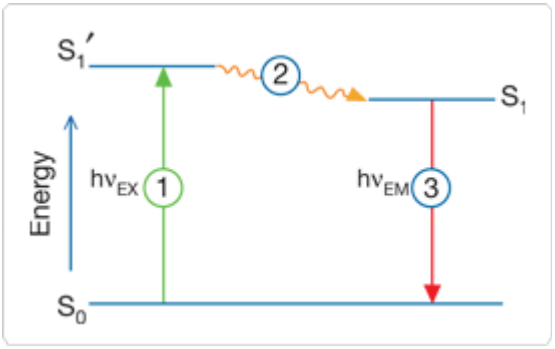


Figure 1: Jablonski diagram illustrating fluorescence process [3]

2.1.2 Application of fluorescence in oil detection

The process of fluorescence is useful for detecting oil spills in water for a number of reasons.

Oil and its products have lower density than water. They are usually hydrophobic. As a result, they tend to stay on the surface of a water body, forming a film.

Since water on its own is not a fluorophore, it does not emit a fluorescent light. However, petroleum components do have a strong fluorescence, and as a result, when an oil film is beneath the sensor, a fluorescent signal can be detected.

False positives are also possible, due to other compounds having similar emission characteristics as oil products. However, the method is sufficiently reliable to detect oil spills.

2.2 ROW oil sensor

The device being produced by LDI AS at the current moment is designated as LDI ROW. It is a non-contact fluorescent oil sensor.

2.2.1 Composition

The device that is currently in production is an “all-in-one” unit. It contains an optical part and electronics, both parts are housed within a single metal casing.

Electronics are implemented as a two-board solution, with two PCBs stacked vertically. One board is responsible for operations and analysis, housing an Atmel microcontroller, as well as an ADC and a large-surface highly sensitive photodiode. The other board is responsible for power supply, but also contains communication drivers.

A powerful ultraviolet LED is contained on a separate board within the same casing.

2.2.2 Principle of operations

The LED emits pulses of light in range of microseconds, governed by a microcontroller. Narrowed by the optics, a ray of UV light irradiates the surface beneath it.

Provided that there is an oil spill below, oil reacts to the light by emitting a fluorescent radiation with a different wavelength. This light is measured by a photodiode. Optical filters allow only the required emission wavelengths to pass through to the photodiode, allowing for separation of the emission signal from both excitation light and major part of parasitic illumination.

2.3 New ROW sensor

The new device will be operating on the same base principles. However, there will be some major differences in functionality and composition.

2.3.1 Major changes

The first and most important difference is that the new device will no longer be “all-in-one” solution. Instead, the new product will be divided into two parts: sensor part and host controller.

The sensor part will only contain data gathering equipment. It will be, same as formerly, contained in a pressurized case filled with a noble gas. However, aside from improved optics, it will only have basic equipment, limited to data acquisition and communications within a fixed channel, with only two options – a wired PLC or a wireless radio link.

The host controller will now be relied upon for performing analysis, making a decision and communicating the data. Aside from the microcontroller and a communication link to the sensor, it will house all supported communications to either a PC, Ethernet or other widely-used interfaces, either wired or wireless.

Another change is that the new device will have four channels instead of one. Therefore, detection reliability and field of application will be improved significantly.

Finally, the new device will come with better interfacing functionality, which would help to avoid the necessity to install additional equipment and blocks by side manufacturers, reducing costs and increasing user-friendliness.

2.3.2 Advantages of two-part composition

There is a number of advantages to utilisation of separate host controller and sensor, which radically changes the functionality and applicability of the device.

First major advantage is reduced difficulty of maintenance. Currently, any in-field maintenance is impossible due to the whole device being enclosed in a gas-tight body. With the new approach, only the data gathering part is inaccessible in field, which decreases costs for maintenance and increases foolproof characteristics and robustness.

Another advantage is that placing a large part of the operations on a separate controller adds more room for functionality expansion. Available computational power of a sensor part microcontroller can be used for processing measurement channels, which was not possible with the former model due to processor limitations. Similarly, a dedicated controller that performs analysis and communications will have more than enough capacity to handle multiple interfaces, data storage and even such functionality as a dedicated web server.

With most of the operations transferred to a host controller, power consumption and reliability of the sensor part will improve. It means that the sensor part can be installed in remote places, which are difficult to access. Due to having a fixed interface between the host controller and the sensor, it is no longer necessary for the sensor to be connected with a thick multi-wire cable. Instead, only a power line is required. Where even that is not possible (in such cases where the sensor is supposed to be installed fully autonomously and some energy gathering method is applied to power the sensor), a wireless link can be used.

Most of competing solutions are using a “host-sensor” model. However, in their case it is usually a relatively short wire that connects the host and the sensor. In the new ROW design, the host controller can be installed at significant distance from the sensor, widening the area of application.

3 Hardware

The hardware part of this job will include a microcontroller and various peripherals, such as communication equipment and safety features.

3.1 Controller

Considering a wide variety of microcontrollers available on the market, RCM6700 series manufactured by Digi Inc was selected. RCM6710 MiniCore, which is to be used in the product in question, aside of Rabbit 6000 microprocessor includes convenient GPIO outputs, external components, additional flash memory and an embedded Ethernet socket.

3.1.1 General ideas

There are some points which should be mentioned when describing a controller element for the device.

Controller types [4]

While talking about controller architecture, there is a slight but noteworthy difference between what is considered to be a microcontroller, a microprocessor, a system-on-chip, a computer-on-module, etc.

Generally, a microprocessor is an integrated circuit which incorporates a CPU on a single chip. In general terms, it should not contain any other equipment, such as memory or interfaces. However, integrated circuits that incorporate a CPU, I/O (including serial ports and other interfaces) and even some RAM, are frequently marketed as microprocessors, usually to make an emphasis on them being part of a larger product (computer-on-module), which makes the difference between microprocessors and other architectures difficult to pinpoint.

The definition of a microcontroller is usually a self-contained computer on a single integrated circuit. These usually have all the required interface drivers and peripherals, such as memory and clock, integrated into the same chip. Microcontrollers are intended for using within an embedded system, usually for fulfilling a single and relatively simple task. As such, the emphasis is made on reduced physical dimensions, low power consumption and cost. Therefore microcontrollers are usually assumed to have lower characteristics than microprocessors and SoCs.

An SoC is a computer contained within a single chip, similarly to a microcontroller. The difference is that an SoC is intended for a wider range of tasks (SoCs often find themselves as parts of miniature personal computers and handheld devices), and as such are supposed to have higher characteristics and larger number of embedded peripherals. However, the division is mostly vague and it is usually the question of how a specific product is marketed.

A computer-on-module, or a system-on-module, is a type of computer designed for embedded use. It is usually a board which contains a microprocessor or a microcontroller, and some peripherals, usually for power, memory or interface management. They are designed as a part of an embedded systems, and as such, are not intended to function on their own. Emphasis is made on compact shape and ability to be integrated into a motherboard.

A single-board computer is a larger version of a computer-on-module, which usually contains a system-on-chip or a microprocessor, along with an expanded number of peripherals and is intended to function as a single unit. As such, it often requires no or few additional peripherals and houses standard connectors, such as a power jack, one or more USB sockets, RJ-45 for network connectivity and HDMI/VGA for video output.

RISC vs CISC [5]

RISC processors, as opposed to CISC, means reduces number of commands and instructions. While CISC targets at as few lines of code as possible, having an independent instruction for many separate operations, RISC will have complex operations divided into multiple elementary ones, which only require one clock cycle to complete. For instance, if CISC instructions handle complete operations, together with

reading and writing memory and registers, RISC would rather have separate instruction for every step (loading a value from the memory location into a corresponding register, performing an operation, writing the result back to memory).

While an advantage of CISC is a smaller code size, which reduces memory consumption and compile time, RISC architecture requires lower number of transistors, and since operand values that are stored in the registers can be reused, allows for better optimization.

3.1.2 Rabbit 6000 processor [6]

Rabbit 6000 is the latest addition to a line of Rabbit microprocessors, which started with Rabbit 2000. It was created by Rabbit Semiconductors, and is now being produced by Digi International, Inc. The processor has been designed for use in single-board computers and embedded systems, dedicating great attention to network connectivity and communications.

Rabbit 6000 is a quite powerful 16-bit processor with a C-optimized instruction set, running at 200MHz and producing low EMI.

This microprocessor also includes many features not commonly found on products of given class. Rabbit 6000 provides 16 channels of DMA, six independent configurable serial ports, a two-channel quadrature decoder, four PWM outputs and an I^2C port capable of 400kbit/s rate and 10-bit addressing.

Rabbit 6000 includes eight 8-bit parallel ports. Six serial ports are available, all configurable as asynchronous. Four of them can be used as clocked serial (SPI). For parallel ports one could configure such properties as embedded pull-up and pull-down resistors, drive strength, direction, et cetera.

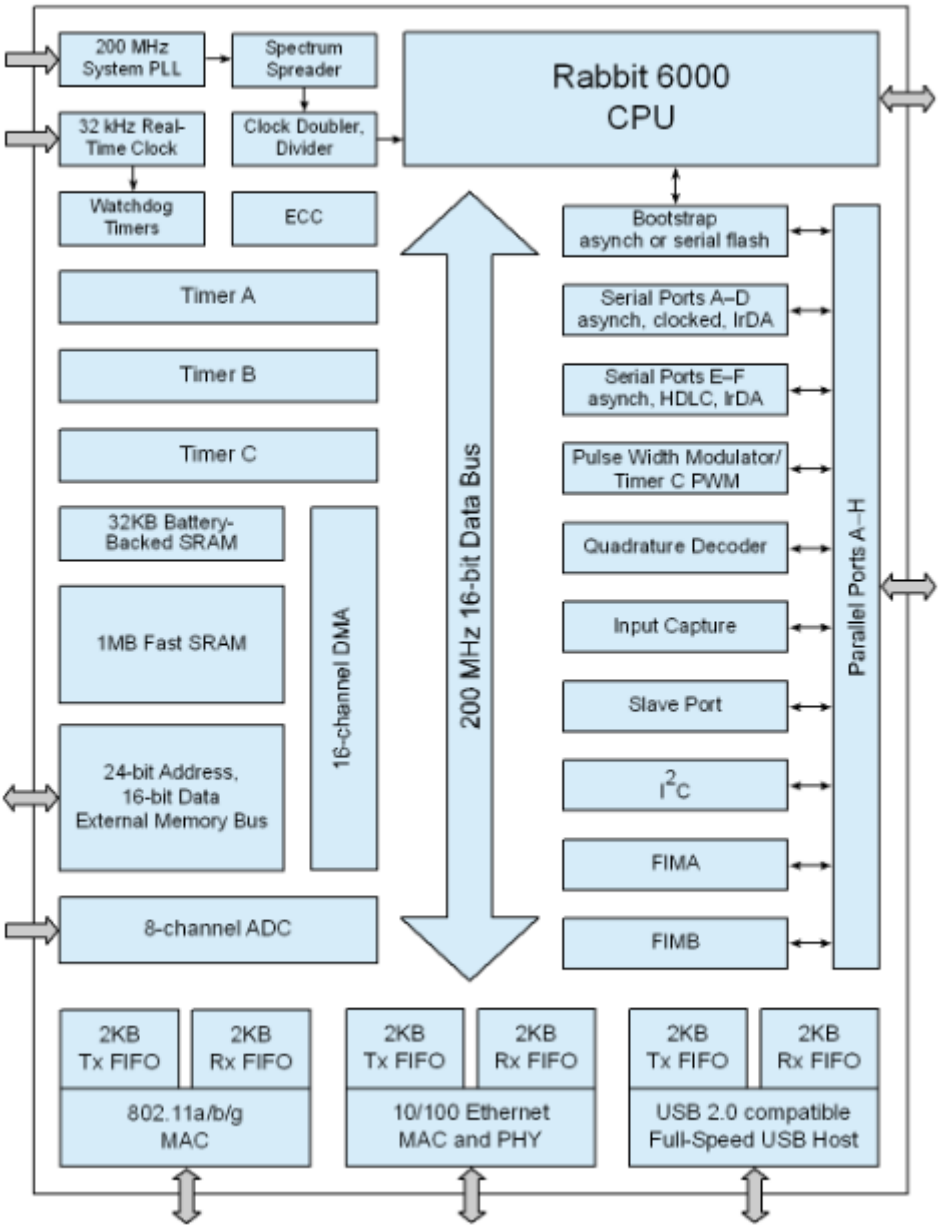


Figure 2: Rabbit 6000 block diagram [6]

Rabbit 6000 has three separate timer systems. Timer A consists of twelve 8-bit counters with a programmed time constant. Timer B has a 10-bit counter, two match registers and two step registers. Timer C is a 16-bit counter. It contains eight match registers, which provides support for up to four PWM or quadrature signals.

The processor also includes a 12-bit, 8-channel analog-digital converter (ADC), capable of performing up to 10^6 samples per second, based on either internal or external clock.

Rabbit 6000 contains 1MB of RAM and 32kB of battery-backed Static RAM (SRAM). They both can be enabled in 8-bit or 16-bit mode. Rabbit 6000 also supports 8-bit and 16-bit external memory devices (Flash or SRAM).

Another feature of this processor is an ability to prioritize interrupts. The Rabbit 6000 can operate at four priority levels, with 0 being a standard operating level and 3 assumed to be highest priority. Current priority level is stored in a dedicate register. Whenever an interrupt occurs, it will be handled only if the current priority level is lower than one of the pending interrupt. Therefore even while an interrupt is being handled, another interrupt can occur and take control, assuming that it has a higher priority. On the other hand, even a high-priority interrupt can be blocked if the processor is at the highest priority level already. This system allows for more efficient handling of interrupts, allowing to let more important events be handled even if a less important interrupt has already been called.

Rabbit 6000 has also eight external interrupt vectors. Each vector can be set to trigger on a rising edge, a falling edge and both edges. The signal has to be present for at least three clock cycles to be detected.

Many functional registers on Rabbit 6000 are one-way, either read-only or write-only. That means a value cannot be read from a register once it is written. To bypass that issue, shadow registers are present, which store the current value of a register to be accessed by a user.

3.1.3 RCM6700 MiniCore series [7]

RCM6700 series is a microcontroller module that incorporates Rabbit 6000 processor. It comes in a Mini PCI Express form-factor. While that means a host PCB has to have a special slot for the MiniCore, the module can be removed for maintenance or replacement and installed back with relative ease.

MiniCore includes a Rabbit 6000 processor, with its I/O pins routed to the PCI interface. It also contains external flash memory, capacity depending on a specific model, a watchdog supervisor, power management hardware and an optional RJ-45 socket. All boards have Ethernet support, either by pins on the GPIO or as a pre-installed socket on the MiniCore board.

Figure 3: RCM6700 MiniCore module

RCM6700 Series consists of four devices that demonstrate different price and functionality. They are all based on the same Rabbit 6000 processor, but the boards, and consequently, peripherals vary.

- **RCM6700**
    - This is a basic model. It has 1MB of serial flash and no additional interfaces except for the 52-pin Mini PCI Express. Of the four devices, it has the lowest power requirements and price.
- **RCM6710**
    - This device is essentially an RCM6700 with an RJ-45 socket pre-installed on the board, which removes the necessity to include it in a board design.

- **RCM6750**
    - This is an upgraded version of RCM6700. It has 4MB of serial flash-memory, 1MB of external SRAM in additional to internal memory. Similarly to RCM6700, it features no RJ-45 socket pre-installed.
- **RCM6760**
    - This version is an RCM6750 with an addition of RJ-45 socket. Of all the model range, it is the most expensive one and requires more power than the others.

There is an optional Development Kit that can be ordered with the module. The kit includes an RCM6700 together with an Interface board and a Prototyping board.

The Interface board is essentially a motherboard. It has a socket for the RCM6700 MiniCore, and various interfaces such as jumpers, LEDs, a +5V power socket, Mini-USB socket and an RJ-45 Ethernet socket, as well as a stacking connector to install extensions, which can also be used to access GPIO of the microcontroller. On hardware part, it includes a regulated power supply to provide +3.3V to RCM6700, USB and RJ-45 drivers and other peripherals. The Interface board also contains a reset switch and a user-configurable button.

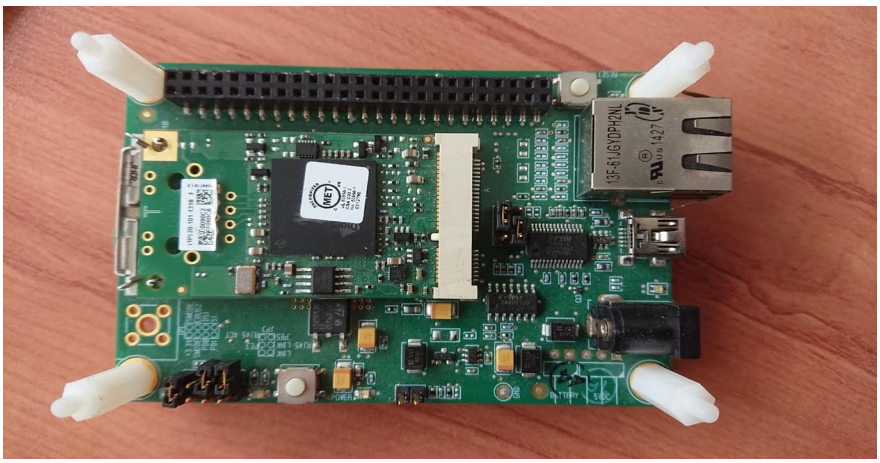


Figure 4: RCM6700 Development Kit Interface board

The Prototyping board features a stack connector to install extensions below and above (the board is intended to be installed on top of the Interface board), and a prototyping area for through-hole and SMD components. Availability of the Prototyping helps to design and debug simple circuits that are intended to work with RCM6700.

An extended version of the Development Kit exists, called Deluxe Development Kit. Aside from the components of a regular kit, it contains a Serial Communications board and a Digital I/O board. Serial Communications board provides interfaces for serial communications and exploitation of available six serial ports of Rabbit 6000. For instance, it includes RS-232 driver. Digital I/O board contains various user interaction components such as LEDs, buttons and configuration jumpers. These two boards are used in sample programs to demonstrate the capabilities of RCM6700.

3.1.4 Alternatives

For the task given, there are numerous alternatives that can be used as a host controller. These include not only industrial microcontrollers, but also SOCs and dedicated off-the-shelf PCs.

Personal Computers

A simple compact computer can be used to work as a host controller. Examples may include Intel NUC Mini-PC or MSI Trident series. They are usually compact enough to fit into an equipment cabinet or electrical enclosure, and cheap enough compared to the end-device price, starting from 100-200€. Wide variety of software and fully-functional OS decrease the worktime needed to implement various aspects, such as communications, and they are capable of working off-the shelf, removing need to develop components and housings.

However, a dedicated PC as a host controller has disadvantages that make it inefficient and sometimes inadequate to the task. First of all, compact PCs are usually intended for home or office use, and as such, do not have necessary robustness to work in an industrial environment. In addition, they have excessive capabilities, functionality and equipment. And while that is not essentially a disadvantage, this means their price is higher compared to some alternatives, and the power usage may be too high. Besides, such PCs are not extremely flexible on the hardware side, limiting their usefulness when

a non-standard connectivity is required. For instance, adding an interface other than USB or Ethernet (such as RS-485) would require an adapter.

Figure 5: Intel NUC Mini-PC

Industrial computers

These devices are also fully-functional computers with all the accessories and components. They also have a widespread OS, such as Windows or Linux Ubuntu, thus demonstrating the advantages a regular compact PC has. However, they are meant for industrial use. As such, they are much more robust and better tolerate industrial or field environmental conditions. They usually have no cooling fan and often have an SSD instead of a mechanical hard drive, as well as industry-rated components, which means they require less maintenance and are less prone to malfunctions. Besides, as industrial equipment they often have interfaces typical for industrial use, such as RS-232.

However, compared to regular PCs they are often more expensive, usually from 400€ upwards. Besides, aside from most used interfaces, the same inflexibility of PCs on hardware part applies to them as well. Their power consumption is, though lower than that of a PC, still relatively high.

A common disadvantage to both industrial and personal computers is that they usually have a non-real-time OS. Installing a real-time OS on a specific computer is often

difficult. While oil pollution monitoring is not an application that requires extremely small response times, that might still be an issue.

Single-Board Computers

An alternative to a ready-to-use computer would be an embedded Single-Board-Computer (SBC) board. A SBC is a complete computer built on a single circuit board. That includes all the necessary components, such as a microprocessor, RAM, flash memory, input-output drivers (usually USB, Ethernet). Usually those are built utilizing a SoC concept. They frequently feature a customized version of a widespread OS (usually modifications of Linux).

While some industrial and personal computers are built as SBCs, there are systems specifically intended for embedded applications, and as such, offer greater flexibility and configurability. They often include some sort of GPIO, common interfaces like UART, I2C, SPI, and various expansion boards. However, such flexibility means that they usually come as a "barebone", requiring development of additional components (both electronic and mechanical) to create a final product.

Raspberry Pi

A well-known SBC is Raspberry Pi series. It was initially created as a cheap computer to promote learning of computer sciences and programming, especially for developing countries. However, the flexibility and functionality of the computer have lead to a high demand outside of target markets.

Although this computer is not rated and was not originally intended for industrial applications, it has found its place in home and industrial automation and even commercial products, which makes it a viable alternative for the given task.

Raspberry Pi has several models available. They all vary in price and capabilities, starting from Pi Zero, which only has the most basic characteristics and functionality, and up till Model B+, which has the best hardware among the range. There has been three generations so far, every of which added some new functionality and expanded the existing hardware. Several offspring design by side manufacturers also exist.

On hardware level, Raspberry Pi is a based on Broadcom system-on-chip that includes an ARM processor and a GPU. The latest generation, Raspberry Pi 3, uses BCM2837 chip which contains a quad-core ARM Cortex A53 cluster, which runs at 1.2GHz. [8]

Raspberry Pi also contains from 256MB (earlier versions) to 1024MB RAM. It has no hard drive or other integrated data storage, but has a slot for an SD-card (MicroSD on later models). An SD-card will serve as on-board memory, storing both firmware (OS) and user files such as documents or programs.

Raspberry Pi has a vast number of outputs and interfaces, which makes plugging in peripherals easy. Raspberry Pi models 2 and 3 have four USB ports implemented using an integrated LAN9514 USB hub. They also have an HDMI output to connect a display, RJ45 Ethernet socket, and in case of Raspberry Pi 3, both WiFi and Bluetooth support. All models have a GPIO pin header (26 pins in earlier models, 40 pins in later ones), and serial port hardware support (UART, SPI, I^2C) on all models except Pi Zero [9] .

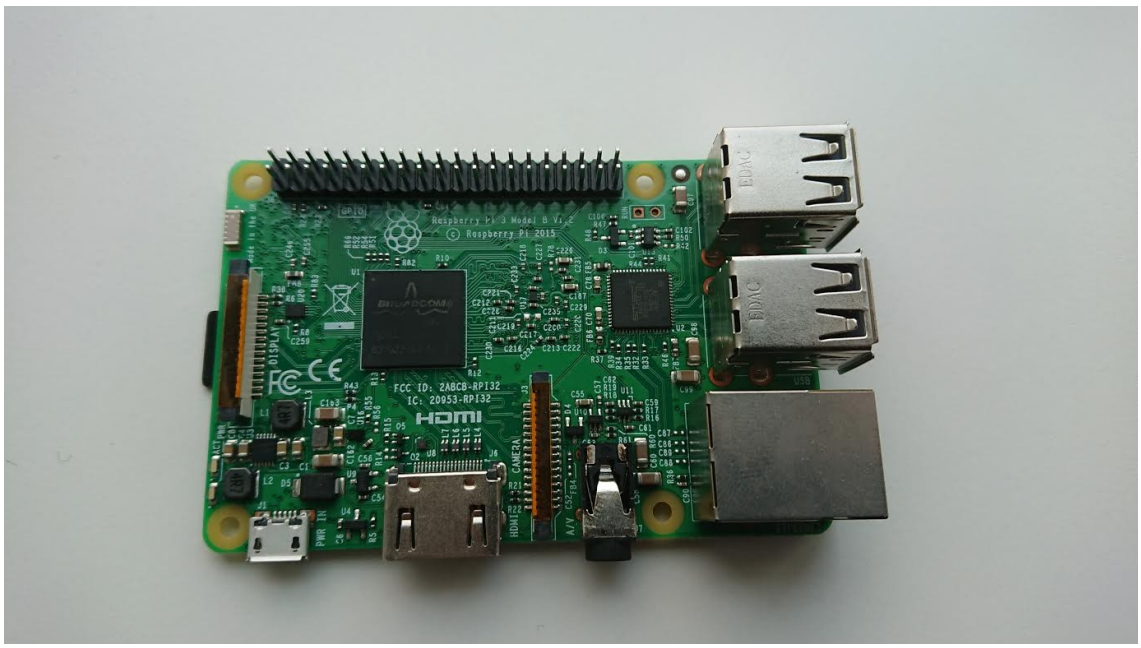


Figure 6: Raspberry Pi 3 Model B

A major advantage of Raspberry is its low price. A Raspberry Pi 3 Model B can be purchased for slightly more than 30€, which is extremely low price compared to alternatives. And it has abundant “off-the-shelf” functionality, starting from USB support, as well as major interfaces and a fully controllable GPIO. Besides, a wide

range of extensions exists for Raspberry Pi. A range of operating systems can be installed on Raspberry Pi, although a recommended choice is Raspbian, a Linux-based OS customized specifically for this computer. This simplifies development of user interface, web services and communications.

On a negative side, Raspberry Pi has not been initially intended as an industrial computer. And while there have been use cases of Raspberry being used in automation and consumer products, its reliability and endurance can be significantly lower that that of alternatives specifically designed for industrial and embedded applications.

Another drawback is that although Raspberry Pi offers higher flexibility in terms of interfaces than a PC, there are still limitations. Raspberry Pi does not have any industrial-grade interfaces, such as RS-485 or 4/20mA current loop, and has only one hardware-supported UART. That means some extensions are to be developed or designed, negating its advantage of being a complete product.

Besides, a non-dedicated OS, especially the one developed on a non-commercial basis, may be less reliable than a dedicated code, and gives less control over its functionality.

Microcontrollers

Microcontrollers are small computers, usually implemented within a single circuit. They are intended for fulfilling a limited number of tasks, usually as embedded system for industrial control or consumer products. As such, they are less sophisticated than system-on-chip or other computers. They usually consist of a microprocessor, small amount of RAM (compared to other computers) and I/O peripherals.

They are optimised for embedded systems, low power consumption and compact design. Therefore they rarely have high computational power or memory capacity.

The main feature of a microcontroller (and an SoC) is that they come as components for an embedded system. It means that while it is not an off-the-shelf product and requires development and manufacture of all the required components, it gives greater control and flexibility to the developer. Besides, microcontrollers are usually cheaper than complete devices.

Prominent microcontroller branches today are Atmel AVR microcontrollers and various products based on ARM Cortex-M processor series, produced by such manufacturers as NXP, Texas Instruments, Toshiba, and others. Other solutions exist today as well, such as Blackfin processor family, made by Analog Devices, or PIC by Microchip.

ARM Cortex [10]

ARM (initially Acorn RISC Machines, later re-branded as Advanced RISC Machines) is an architecture of RISC microprocessors intended for embedded systems and portable devices, such as smartphones, handheld computers and system-on-chips.

ARM is not a processor model or manufacturer, but rather an architecture as a general. The ARM Holding is not producing the processors themselves, but rather licenses the design to various manufacturers that produce microcontrollers or SoCs.

There are currently three base Cortex designs. Cortex-A is a performance version, optimized to support full operating systems, such as Linux or Windows. As such, it is mostly intended for use in computer devices, such as laptops, netbooks or smartphones. Cortex-R is a version intended for real-time operations, where emphasis is made on quickness of response and high speed of operations. As such, it is often used in devices that rely on real-time performance, such as modems. Finally, Cortex-M is a version intended for microcontrollers. As such, it exhibits low power consumption, smaller size and lower cost. It is mostly used in embedded systems.

Cortex-M series consists of seven models distributed into three groups by their functionality and price. [11] Cortex-M0, Cortex-M0+ and Cortex-M23 belong to the first group, featuring lowest power consumption and smallest area. The second group consists of Cortex-M3, Cortex-M4 and Cortex-M33 and is a balance between performance and cost. The two latter processors feature a DSP for faster and more power-efficient handling of signals and signal control applications. The last group consists of a single member, Cortex-M7. It features the higher performance, at the cost of higher energy consumption and area. M7 model also includes a DSP.

Cortex-M processors support 8-bit, 16-bit and 32-bit operations. Performance and memory characteristics largely depend on the manufacturer of a specific

microcontroller. The range of ARM-based controllers is wide and capabilities vary to a large extent.

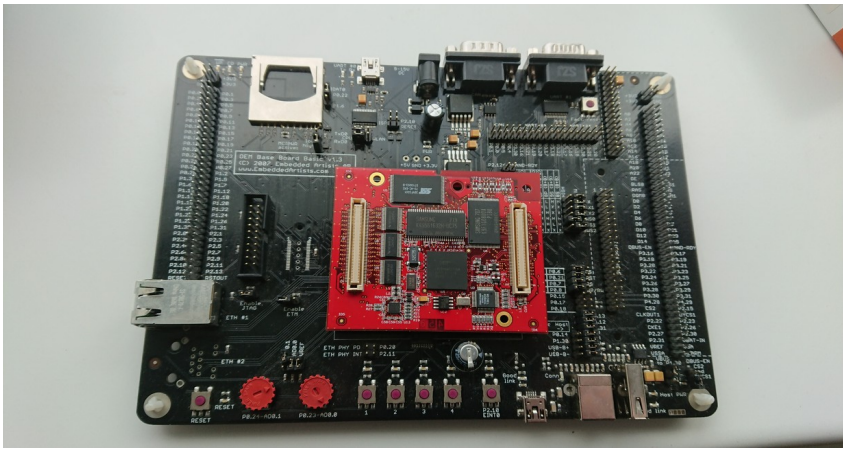


Figure 7: NXP LPC2468 (based on an older ARM7 processor) module with Keil MCB2360U development kit

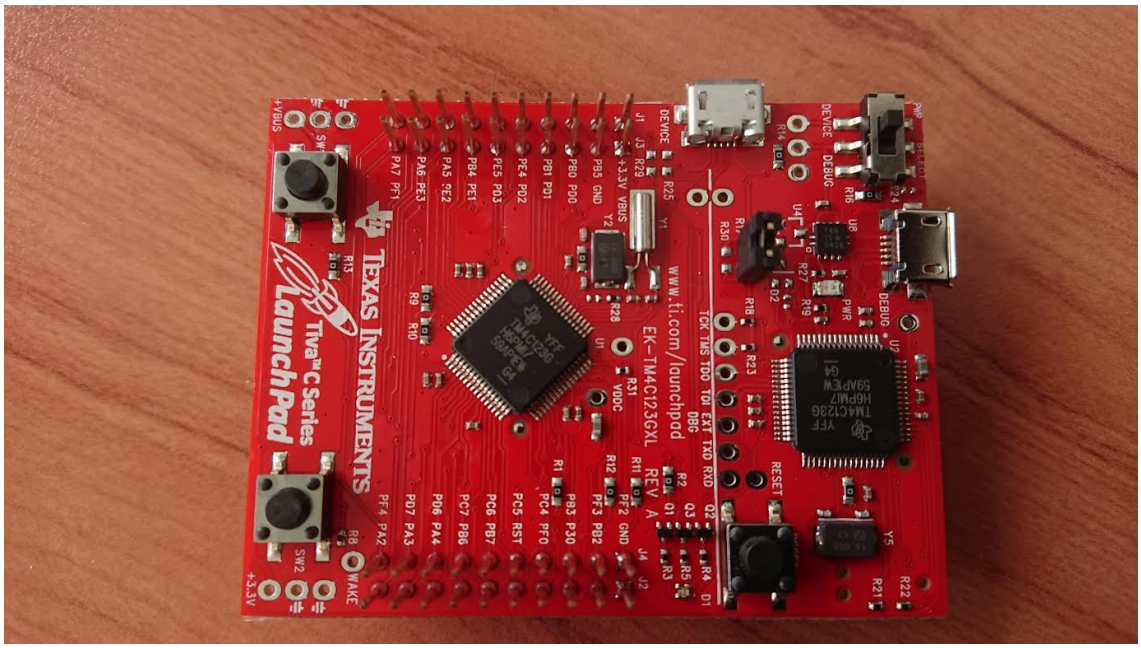


Figure 8: Texas Instruments LaunchPad with a TMC123G controller based on ARM Cortex-M4 series. This LaunchPad contains an additional microcontroller for programming and debugging.

Atmel AVR

AVR series was conceived by two students of Norvegian University of Science and Technology, Alf Bogen and Veggard Wollan. Most microcontrollers of its time had only EPROM to store program data. This kind of memory was not electronically reversible, and once written, could only be reprogrammed after the previous data was cleared by exposure to ultraviolet light. That required expensive quartz windows in the design of a microcontroller, which increased the cost, and made a debugging process inefficient, because erasing memory under ultraviolet radiation was a time-consuming process. AVR microcontroller design involved use of flash memory to store program data, therefore increasing production and debugging speed and decreasing the costs. [12]

Atmel AVR range consists of four families. These families vary in their cost and characteristics. [13]

Atmel tinyAVR microcontrollers are 8-bit devices optimized for power efficiency, ease of use and small physical dimensions. These chips have an integrated ADC, flash memory, on-chip debug and other features.

Atmel megaAVR controllers feature higher performance, increased memory capacity and various sets of peripherals to satisfy user's needs, including USB, LCD controllers, CAN, LIN drivers, et cetera.

Atmel AVR XMEGA is a series of 8/16bit microcontrollers that feature higher functionality and performance characteristics than tinyAVR and megaAVR.

Atmel AVR UC3 is a 32-bit microcontroller family, which is considered the flagship of the model range. It features DSP support, SRAM, DMA controller and other features not found in lesser models.

AVR controllers are RISC devices. They support various programming interfaces, such as ISP (In-System Programming, an on-circuit programming interface performed through SPI bus), PDI (Program and Debug Interface, a proprietary interface for programming and debugging XMEGA controllers), High Voltage Serial, etc.

Atmel offers a number of development tools and starting kits for evaluation, learning, initial design and debugging. The chips themselves come in a variety of packages, depending on the required functionality and the number of GPIO pins.

Figure 9: Atmel AT89LP51 microcontroller (8051 architecture), the one that is currently used in ROW

3.1.5 Considerations for selecting RCM6700

A decision to use RCM6700 module for the host controller was made on grounds of several aspects. Important considerations to be taken into account are the specifics of LDI AS and the device to be produced.

One of those specifics is that the company is relatively small. That means that separation of tasks between employees is minimal. Consequently, LDI cannot dedicate as much manpower and resources as larger companies. Therefore, whenever a solution can be reused or simplified instead of being thoroughly reworked, it means large savings on time for the company, making products that have more in-built features and tools a priority to use.

Another aspect is that the final device that is being developed, oil spill sensor, is not a product to be manufactured in high numbers. Production rate is usually limited to several devices per month. Therefore, the cost of development is more pronounced in the final device prime cost, which should be kept low if possible for obvious reasons. And this means that if possible, spendings on instrumentation, proprietary software or other costs involved in design and development should be minimized.

Considering all that aspects, RCM6700 series module has been chosen for the host controller, for the following reasons.

First of all, Rabbit 6000 microprocessor, which is part of the RCM6700 series module, is highly advanced, compared to the competition of its price range. Its performance characteristics are superior to many microcontrollers. The availability of integrated functionality, such as multiple serial ports, various interfaces, including an on-board Ethernet socket, and additional tools, make the workflow quicker and put less strain on development of software and additional hardware. A form factor also means easier placement on a motherboard and disassembly when necessary.

Another important aspect is that RCM6700 comes with a number of development tools. The development kit is a universal and functional platform for debugging and testing hardware and software solutions. Since the development kit comes with a number of extensions, which allow for various functionality of the controller to be tested, the need for special hardware design for development stage in negated.

RCM6700 also comes with a free IDE, Dynamic C, which is optimized for designing software and programming a Rabbit microprocessor. Although the Dynamic C IDE might lack some functionality compared to IDEs produced by larger companies, such as ARM Keil, it is sufficiently powerful. The fact that it is available for free is an advantage considering the aforementioned specifics of the company and the product.

Rabbit 6000 processor instruction set is optimized for C programming language, which is the language used in the workflow. Besides, software and firmware for earlier LDI products has been written in either C or its derivatives (C++, C#), which means previously developed functionality can be reused with little adaptation.

The processor uses a custom dialect of C language, which is called Dynamic C. This dialect contains additions to standard C language that help to better use the functionality of Rabbit 6000, simplifying some tasks that would be harder to implement otherwise. Both Dynamic C and a large number of macros supported by the compiler make such problems as parallel processing, multitasking and communications more efficient and easier to implement.

Dynamic C software package also comes with a vast variety of libraries, tuned for the latest series of Rabbit microprocessor, supporting the functionality integrated into the processor, such as quadrature decoders, PWM, serial communication on such interfaces as I^2C and SPI, TCP and UDP communications, and many more. There is a large collection of examples provided with the software pack, that contain use cases of those libraries and peculiarities of Dynamic C.

Such availability of ready-to-use components and examples that demonstrate how to implement specific tasks with these components means that less time and resources should be spent on research and development. This results in shorter time-to-market and development cost.

There are some disadvantages, however, that should be taken into account while designing an RCM6700-based device.

First of all, while the form-factor of the module is convenient for easy removal and replacement, initial placement of the module on a new board is more complicated. A Mini PCI Express socket should be installed on a motherboard, together with clamps that support the module from the other side. The combination of two requires high precision in installation.

Another disadvantage concerns the language that is used in the development. Dynamic C, a dialect of C language, is mostly compatible with standard C, but there are some differences. Some of the standard aspects are redefined, and there are some limitations that are not present in C language. Besides, some of the macros definition might differ from what other widespread compilers use. This makes adopting a program written for another compiler, or even a program written for an earlier version of Dynamic C, more difficult and time consuming.

There are some nuances in the functionality of Rabbit 6000 processor. For instance, most periphery registers are write-only. That means their value cannot be read by the program. This limitation is usually negated by use of “shadow registers” (variables that are assigned the same value as the corresponding register and that can be read to find out which value the register stores).

Finally, Rabbit 6000 and the RCM6700 series are not as popular among developers as other widespread controllers, such as AVR or ARM. This means that, while a large library and toolkit is available from the manufacturer, the information and use cases from other users and side developers is scarce, compared to more popular microcontrollers that have a vast amount of solutions available by amateur users and side developers alike.

3.2 Power-Line Communications

One of the two ways the host controller will be using to communicate with the sensor is a power-line communication. This concept describes using power wires to transmit data, thus removing need for additional communication wires.

3.2.1 Advantages of PLC

In an oil spill sensor, an important task for the communication line between the sensor and the host controller is to use minimal amount of wires. An oil sensor is often installed in remote places, sometimes in areas with an explosive atmosphere (falling under ATEX directive), or other hazardous environments. For this kind of purpose, a multi-wire cable that can accommodate both power and communication lines (for instance, RS-485 interface) might be extremely expensive and difficult to use.

The amount of data transferred between the sensor and the host controller is not extremely high. Therefore, high transmission speeds are usually not required in a regular work. However, for some situations high throughputs are necessary. For instance, remotely updating the sensor firmware takes requires transmitting a large amount of data over short time. Therefore a line should be capable of maintaining higher speeds when needed.

It is not always possible to use wireless connection. Usually, ISM band does not require licensing and can be used freely. However, in some areas there are some special frequency regulations on the site that make using wireless link impossible. Aside from that, wireless communications may perform poorly in areas with high electromagnetic disturbance or when a direct line of sight is blocked.

In this cases, a PLC link is a logical choice. It leaves only two power wires in a cable, with communications being conducted over power lines. As such, cheaper and more compact cables can be used.

There are some advantages of PLC to consider. First of all, as with most kinds of communications, a balance should be achieved between operational range and channel throughput. And in case of PLC, range and throughput are both comparatively low. With a scenario that does not require high throughput, range can be significantly extended. However, it is still lower than when using a dedicated wire channel, such as RS-485.

Another issue to take into account is a high demand to the quality of a power line. Disturbances in a power level, which are too small to be taken into account for the purpose of powering the device, and as such usually not compensated, can become a problem for PLC, increasing noise level and consequently lowering both range and throughput. In case of the device being developed, this problem might not be so pronounced since the signal will be transmitted over a dedicated power line from the host to the sensor, not a mains AC line.

PLCs are usually divided into two distinct groups. Narrowband PLC is for a long-range low-throughput communication. It works at lower frequencies (up to 500kHz), and consequently, lower data rates (hundreds of kbps). It has a longer range (up to several kilometres), and is often used in smart grid and remote sensor solutions.

Broadband PLC works at frequencies of several to several hundred MHz. It allows data rates of up to hundreds of Mbps over much shorter range (usually within a single office or floor). It is used in home networking or broadband access (Ethernet over power). [14]

In our task, narrowband PLC should be sufficient for communicating between the sensor and the host controller. The channel does not require Ethernet-grade throughput, however, larger range would be an important benefit.

3.2.2 SIG60 PLC transceiver [15]

A SIG60 is a power line transceiver for UART/LIN interface designed and produced by an Israeli company Yamar Electronics Ltd. It is a cost-effective solution for digital communication over power lines. SIG60 uses a unique multiplex signalling technology that compensates for noisy environment of power lines. SIG60 is capable of speeds up to 115.2Kbps.

SIG60 operates on AC/DC power lines and contains a UART host interface, which simplifies its connection to a microcontroller. It supports multiple networks working over the same power line, as well as multiple devices connected to a single host device in a bus topology.

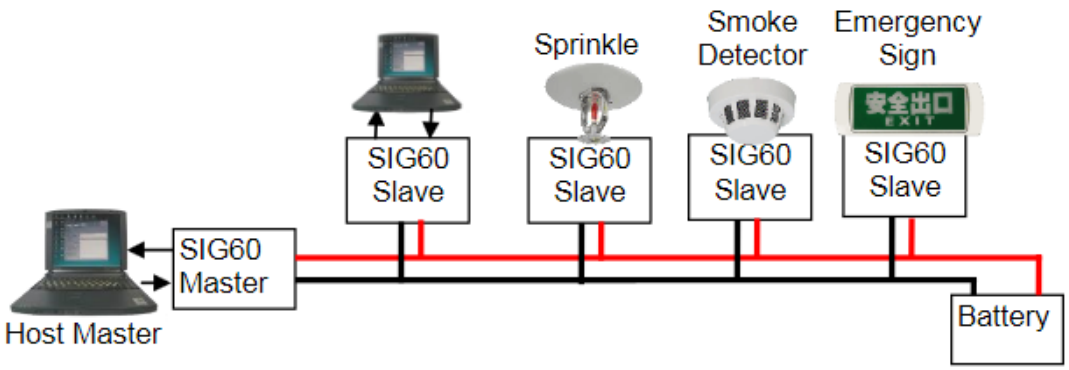


Figure 10: Example of SIG60 in fire protection network [15]

SIG60 features a command mode which can be used to control the operations. In command mode, UART interface can be used to write to internal registers of SIG60. There are two internal registers that allow to select a channel to be used, define frequency of those channels, change bit rate and modify other parameters.

SIG60 can use two channels. In case of a congested or noisy environment, frequency hopping is supported. However, that might not be required in a link between the oil sensor and the host controller because number of devices in the network is low.

3.2.3 Design

SIG60 datasheet features a pinout with description of pin functions and requrements.

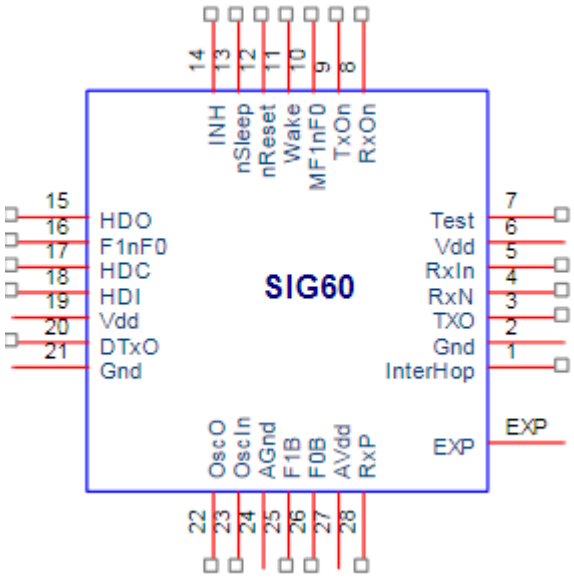


Figure 11: SIG60 pinout [15]

SIG60 requires a number of external components to be added to the design in order for it to work. Principal schematic, as well as a more detailed typical application schematic, is provided in a datasheet.

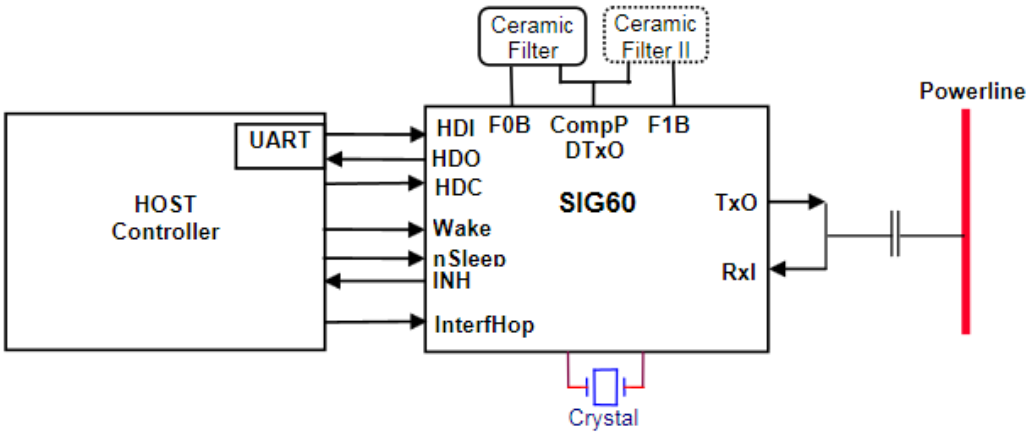


Figure 12: SIG60 interfacing and external components [15]

As can be seen from the block scheme, the main requirements to be added are an oscillator crystal and a ceramic filter for every channel in use. SIG60 is designed to operate with a 4MHz crystal which is to be connected to pins 23 (OscO) and 24 (OscIn) of the IC. Both pins may also be tied to ground through capacitors, according to crystal manufacturer recommendations.

Both available channels require a separate ceramic filter to be installed. The frequency of this filter defines bandwidth of the channel. Since the host controller will use only one channel to communicate with the sensor, and frequency hopping is not required, it is possible to install only one filter. According to the datasheet, a 6.5MHz filter allows for up to 57.6 kbps, which is sufficient for the task.

Filter should be tied to the positive pin input signal (pin 28, RxP) and the corresponding filter pin (in our case it is pin 26, F1B) though a capacitor.

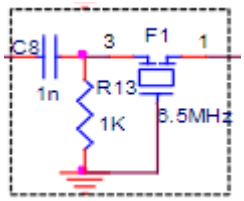


Figure 13: Schematics for a 6.5MHz filter [15]

RxP pin is also tied to RxN pin (negative input signal) through a 470 ohm resistor.

Pins 15 (HDO, UART Data Out), 17 (HDC, a mode switching pin, which is to be pulled low to access SIG60 command mode, and kept high otherwise) and 18 (HDI, UART Data In) are to be tied to designated pins of the microcontroller. For pins 17 and 18, since they are input pins, a 100 kOhm pull-up should be added.

3.3 Sensor-host wireless communication

Sometimes using power-line communication between the host and the sensor is not possible, or is difficult. PLC implies that the sensor is powered directly from the host controller, which might not be an optimal solution in some situation. There are numerous cases when the sensor is powered independently, for instan For these cases, there is an option to use wireless link

3.3.1 Considerations

There is a wide variety of protocols and solutions on the market today, that can be used for a sensor-host wireless link. They differ in price, characteristics, functionality, et cetera.

Since high security is not demanded in given application, encryption and security features are not of an essence in the link.

Baudrates required are not high. Since the data comes in relatively infrequently and in small batches, no baudrates higher than 50-100kbps are required.

Low power consumption is necessary when it comes to autonomously planted devices, where energy gathering is applied. For instance, devices planted on buoys cannot receive power via a cable, and therefore should be powered by some autonomous source, such as a solar panel with a battery. For such applications, power consumption of the radio module must be kept as low as possible, and not exceed 100mA.

Although the sensor is usually not located far from the host controller, distance can still be an issue. In addition, the device is intended to be installed in industrial environments, such as ports or factories. In such locations, there might be significant EM interference, as well as various obstacles blocking the line of sight between the sensor and the host controller. Therefore, a range of the radio module should be sufficient, and the link should be able to withstand external factors.

To avoid legal issues, and to save on running costs, the radio module should operate in license-free frequency band. Due to worldwide market of the devices, this frequency band should be recognised globally.

3.3.2 License-free frequencies

Although most frequencies are being highly regulated, and frequencies being allocated among users, there are some frequency ranges which are free for use without licensing as long as certain conditions are met.

The most prominent of those is an Industrial-Medical-Scientific (ISM) band. The following frequencies fall under ISM regulations: [16]

Table 1: ISM band frequency allocation

Frequency Range		Centre Frequency	Availability	RR No.
6765 kHz	6785 kHz	6780 kHz	Worldwide	5.138
13553 kHz	13567 kHz	13560 kHz	Worldwide	5.150
26957 kHz	27283 kHz	27120 kHz	Worldwide	5.150
40.66 MHz	40.70 MHz	40.68 MHz	Worldwide	5.150
433.05 MHz	434.79 MHz	433.92 MHz	Region 1	5.138
902 MHz	928 MHz	915 MHz	Region 2	5.150
2400 MHz	2500 MHz	2450 MHz	Worldwide	5.150
5725 MHz	5875 MHz	5800 MHz	Worldwide	5.150
24 GHz	24.25 GHz	24.125 GHz	Worldwide	5.150
61 GHz	61.5 GHz	61.25 GHz	Worldwide	5.138
122 GHz	123 GHz	122.5 GHz	Worldwide	5.138
244 GHz	246 GHz	245 GHz	Worldwide	5.138

Those bands falling under ITU Radio Regulation (RR) No. 5.138 are frequency bands designated for ISM applications. They require a special authorization to be used from a local administration in agreement with other administrations, whose radiocommunication services might be affected.

The bands that full under RR No. 5.150 are also designated for ISM use. These bands, however, require no authorization and can be used freely. Radiocommunication services using these bands must accept harmful interference caused by ISM applications.

For frequency allocations the world has been divided into three Regions. Region 1 means Europe and Africa, Region 2 – Americas and Region 3 is Asia and Australia.

Short-range devices (SRDs) falling under certain criteria (limited to tens of milliwatts output level) can operate license-free on certain frequencies. These are often used for industrial control or amateur communications (model planes, etc.).

Table 2: Bands intended for SRDs

Band (MHz)	**Region**
433	Europe
458	UK
868	Europe
915	USA
2400	Worldwide

3.3.3 Protocols and solutions

There is a large variety of protocols and solutions on market. They differ in range, bandwidth, cost, et cetera. Not every of them is suitable for the task at hands.

WiFi, based on IEEE 802.11 specifications, is a high-throughput low-range (usually tens of metres, but using a directional antenna, several kilometres can be achieved) standard intended for wireless local area networks. WiFi operates at ISM frequencies of 2.4GHz and 5.8GHz, and does not require a license. For an oil sensor, their throughput is highly excessive, and low range limits the usability of the end product.

Bluetooth is a standard initially intended for mobile devices and personal area networks, as a wireless alternative to RS-232. It operates in 2.4 GHz ISM band, employing frequency hopping. Considering its original purpose, it has moderate range, relatively high throughput (up to 50Mbit/s) and low power consumption. Bluetooth includes three classes of devices.

Table 3: Bluetooth Power Classes [18]

Class	Range (m)	Max. power (dBm)	Power control
Class 1	~100 m	20	Mandatory
Class 2	~10 m	4	Optional
Class 3	~0.1-1 m	0	Optional

As with WiFi, high throughput of Bluetooth is excessive for the needs of an oil sensor. Range of up to 100m (in case of Class 1 transceivers) may be higher than that of WiFi, however still insufficient.

ZigBee is a protocol intended for low-cost personal area networks and host-sensor transmissions, such as home automation, wireless switches, or industrial control. It supports working in a mesh, operates on ISM and SRD bands and supports data rates of up to 250kbit/s. ZigBee devices have low power consumption and range of up to 100m.

There is a number of proprietary protocols, developed by manufacturers of radio solutions.

3.3.4 XBee SX overview

XBee SX is a family of low-power RF modules developed by Digi. There are two models available, an XBee SX model for Region 2, operating at 915 MHz frequency band, and XBee SX 868, intended for Region 1 and operating at 868 MHz. Both are otherwise similar in most aspects. During the development stage, XBee SX 868 will be used.

An XBee SX 868 is a successor to the older product, XBee 868LP, being compatible in both pinout and commands to the latter, while allowing for greater flexibility in selecting transmission power and power consumption, as well as software and hardware improvements.

It can be connected to a microcontroller using either UART or SPI. Former will be used in the product in order to maintain simplicity and considering RCM6700's available six serial ports, as well as to maintain compatibility on software level with SIG60.

It requires 3.3V supply to operate, and has an effective range of over 1km with a stock antenna (specification mentions up to 14.5km range with a 2.1dBi antenna). [19]

XBee on-board parameters can be configured either by a microcontroller, using a set of AT commands, or from a PC using a software called XCTU. A modem can be connected to PC either directly through a development kit, using a USB cord, or remotely, from another modem (which should be switched to a certain mode).

Figure 14: XBee 868LP with a Development kit

3.3.5 Considerations for choosing XBee SX series

XBee is not the only available solution on market for wireless communications. But it is a good choice for the oil spill sensor for a number of reasons.

First of all, it operates on license-free bands (868MHz for Europe and 900MHz for the US). Its transmission power is limited to comply with short-range device regulations. Thus, no permission or agreement from any third party is required to use this modem.

While having relatively low power consumption, it can provide sufficiently high range, which is beneficial for the product since the sensor and host controller can be kept further apart.

Its proprietary DigiMesh protocol does not only provide reliable and sufficiently secure link, but also supports working in mesh topology. It means that every device in the mesh can act as a retranslator, and when there are several sensors installed, the operational range is increased multiple times.

The price of a single module is relatively low. While cheaper solutions can be found, they do not usually have the same capabilities as XBee SX series.

The sub-1GHz band, used on XBee SX series modules, is lately often considered obsolete, compared to 2.4GHz and higher ISM bands. However, its longer wavelength provides better handling of obstacles, compared to higher frequencies. In addition, congestion is lower compared to 2.4GHz (which is in use by many protocols and wireless links). Compared to 433MHz ISM, it still provides smaller antenna size and less congestion.[20]

XBee SX series is extremely easy to connect to a microcontroller. It only requires power and UART, bringing the number of required PCB leads or wires to 4 (+3.3V, GND, UART Tx, UART Rx).

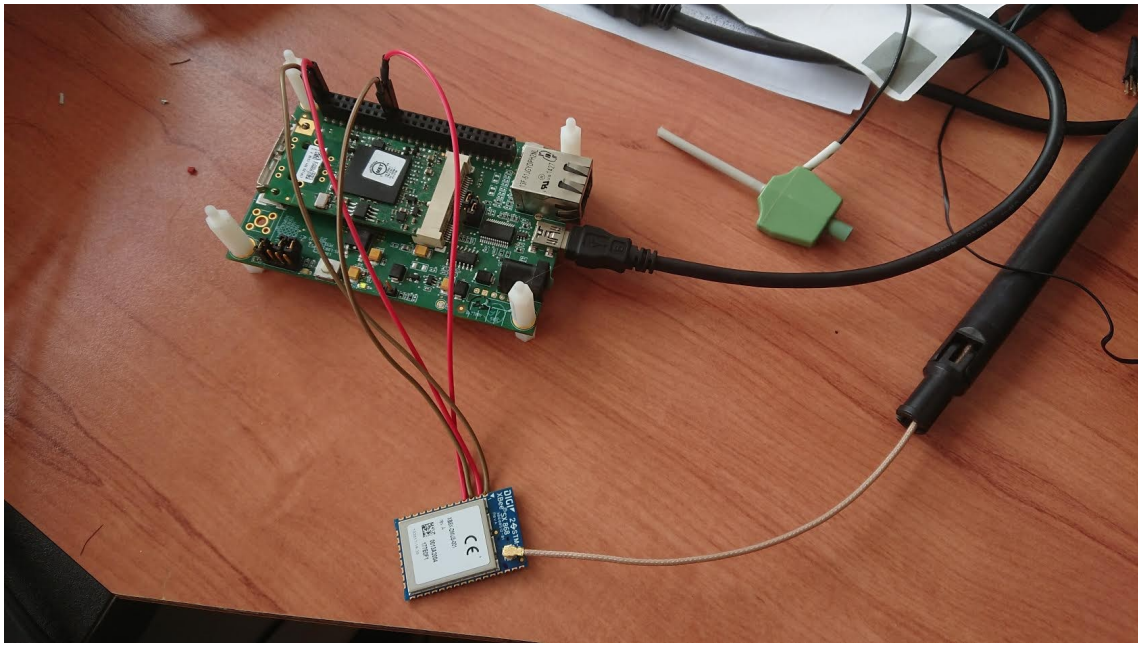


Figure 15: Connecting an XBee SX 868 to an RCM6700 development board for testing

It is easy to configure, either using AT commands, using a PC and a special software XCTU, or remotely, using another modem. There is a huge number of configurable parameters that include output power, baudrate, sleep behaviour, available channels and custom I/O settings. Transparent mode allows for all the serial traffic to be redirected to and from the controller, allowing for “cable replacement” operations.

A baudrate of 80kbps is considered low by today's communication standards, but is adequate for the oil spill sensor.

3.3.6 Range tests

To ensure XBee is capable of providing sufficient reliability in real-life conditions, range tests were conducted using XCTU Range Test tool.

A Range Test involves a local modem sending a number of packets to a remote modem and expecting a reply for each packet. RSSI is measured on both sides and displayed for every packet. Reliability can be judged by how many packets are lost in process.

For testing, two XBee modems were used. A local modem (connected to a PC) was an XBee SX 868 module (Tx power level – 32 mW) with a development kit, a remote modem was XBee 868 LP (Tx power level – 25 mW). In both modems, a Linx Technologies ANT-868 dipole antenna was used. This antenna is intended to be placed on the final product, and features weatherproof and waterproof assembly, as well as rugged design for outdoor installation, an SMA connector and 2m cable. The antenna is matched for 868 MHz frequency and has gain of 0.6 dBi.

Testing was conducted from 4 different locations at different distances. For every test, 10 packets were sent and received.

Figure 16: Range testing local modem locations (blue stars) and remote modem (red star)

Table 4: XBee Range test results

Location	Distance (m)	Obstructions	Avg. RSSI (Loc., dBi)	Avg. RSSI (Rem., dBi)	Packets lost (%)
L1	150	None	-60	-60	0
L2	175	Metal hangars	-75	-75	0
L3	675	Vegetation	-94	-96	0
L4	1100	None	-90	-90	0

As can be concluded from the test results, the signal is reliable with no packet loss up in range of up to at least 1 km with no major obstructions. With major obstructions blocking the line of sight (hangars, buildings) maximum range was seen to be around 300m. Testing of longer range was not necessary since in most cases 1km is enough for usual applications of the sensor. A high-gain antenna can be installed when longer operational range is required.

3.3.7 Connection to a controller

Connecting an XBee SX series modem to a controller only requires power and UART. Handshake signals are supported, but not necessary and will be omitted since data rates are not high enough to cause a problem.

XBee SX is supposed to be powered by 3.3V DC. Since RCM6700 module requires the same voltage, their power source can be shared.

Serial port transmitter (Data Out, DO) and receiver (Data In, DI) can be routed directly to corresponding pins of the microcontroller (Serial port C, PC3 as Rx and PC2 as Tx).

An antenna used will be the same ANT-868 from Linx Technologies, as the one used in testing. Since the antenna has an SMA connector and XBee SX has a U.FL connector, an adapter is required. Another possibility is to use stock 2.1dBi dipole antenna. This one, however, is not waterproof and is only suitable for indoor placement.

3.4 Auxiliary hardware

Aside from major components mentioned above, there is also a number of other components required for the functioning of the host controller.

3.4.1 RS-232 [22]

Many industrial applications demand availability of RS-232 interface. It is a relatively old interface, but still in use by most of industrial equipment, including wireless modems, control and automation equipment, etc. For this purpose, compatibility with RS-232 gives the host controller a major advantage.

RS-232 is a full duplex interface. In the most simple form, it consists of two data wires and a ground wire. Handshake wires may be added to implement flow control, but not necessary. RS-232 operates at voltage levels from (-3..-15)V as “low” (logic 1) to (+3..+15)V as “high” (logic 0). Typically +-12V range is used. By today's standard, with widespread of TTL and CMOS logic, these voltages are rarely present and require an additional power supply or a step-up converter is required, together with voltage inverting.

By standard, baudrate is limited by 20 kbps. This is extremely slow for most of today's applications. Most modern RS-232 products allow for higher baudrates, up to 300kbps, which is also slow compared to Ethernet, but is more adequate to industrial purposes. For an oil detector host controller, such speeds are sufficient.

Another disadvantage of RS-232 is the connector size. DB-9 connectors, which have become de-facto standard in RS-232 communication, are considered to have unnecessarily large dimensions. Some RS-232 products use an RJ45 connector instead, which has more compact shape and can be better fitted into modern devices.

Communication line length is limited by RS-232 standard requiring maximum cable capacitance of 2500 pF. It limits the cable length at typically 20..30 m, depending on a cable used.

There is a huge number of RS-232 drivers on the market. Due to previous experience in the company, ADM2302 driver by Analog Devices will be used.

3.4.2 RS-485

Another widely used industrial standard is RS-485. Unlike RS-232 it features a bus topology, which means several devices can connect to a single communication line.

RS-485 is a relatively outdated standard, but finds its use extensively in industrial applications, due to its simplicity and reliability. RS-485 uses 2 wires, A and B (inverted and non-inverted signal), with another wire (ground) optional. Voltage levels are compared between the two signal lines (A and B), not between a signal wire and a ground. This, together with utilisation of a twisted pair, helps to reduce noise significantly, allowing for much larger distances than RS-232. According to standard, maximum range of RS-485 communication is 1200m, with speeds from 100kbps at long distance (close to maximum) to 35Mbps at short distances (12m). [23]

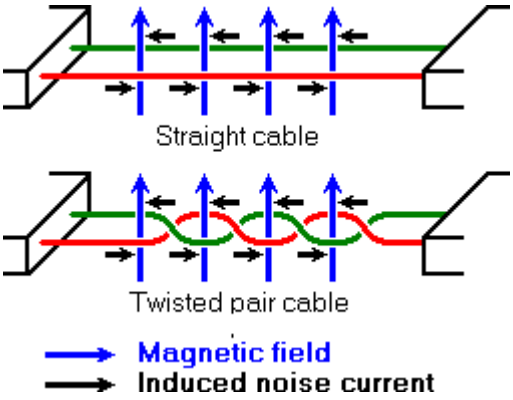


Figure 17: Noise in straight and twisted pair cables [23]

The RS-485 devices can be drivers (transmitters) and receivers. Modern RS-485 devices are often intended to work both ways, as transceivers.

A receiver should recognize a differential voltage between two inverted lines of larger than 200mV as logic 1, and below -200mV as logic 0. Any input between -200mV and +200mV is to be considered indeterminate. A receiver should be able to handle CMR of -7V to +12V, as well as have an input impedance of at least 12 kΩ.

A driver should be capable of delivering at least 1.5V differential voltage in a 54 Ω differential load. 54 Ω is an estimated load of a line with two 120 Ω termination resistors and 32 receivers. A large delta between ±1.5V provided by transmitter and

±200mV sensed by receiver should allow for noise immunity and ability to handle attenuation from long cable. [24]

As shown earlier, the number of devices in the line is limited by a driver's ability to produce enough power to pull the differential voltage to at least 1.5V with a load of 54Ω. This assumes that every receiver in the line has an impedance of 12 kΩ. In case the receivers in the network have a higher impedance, and consequently, lower load, number of the devices can be increased accordingly. Modern devices usually have significantly lower load than required by standard. Therefore a unit is introduced that determines how much strain a device puts on the line, which is called Unit Load (UL). A standard 12kΩ device has a UL of 1. The network is capable of supporting a total of 32 UL. Modern-day devices are often specified as having ¼ UL or ⅛ UL. This means that, for example, a network of ¼ UL receivers can support up to 4*32 = 128 devices. [25]

Terminal resistors can be used at long lines, to avoid formation of a standing wave. They are 120 Ω resistors at every end of the line.

Based on the previous experience at LDI, SP3082 transceiver by Exar was selected.

3.4.3 4-20mA Current loop

An analog current loop is an industry standard that has been in use since 1950s. They are low-cost solutions, relatively immune to noise and are able to carry signals over long distances. For those reasons, they still find use in industrial applications, transmitting data from transducers to instruments.

Current signals are preferred in voltage signals in noisy industrial environments, since they are more resistant to noise and cable losses. Even in a long communication line, current stays the same in every point of the loop.

There is a number of disadvantages that have to be considered. The most important is its power consumption. A loop running at 12V and in its high state of 20mA will consume 240mW, which is much more than most other ways of communication.

4-20mA interface can be used to represent either an analog or digital (three-state) value. In analog mode, a signal can be transmitted in its analog form, using 4-20mA range as a

full scale. In digital mode, a single binary state can be transmitted (for instance, 4 mA may indicate there is no alarm, 20mA indicates presence of an alarm, and 0mA is an indication of a malfunction or a broken wire).

A convenient 4-20mA driver is an Analog Devices AD5412 DAC with a programmable current source and voltage output.[27] It can provide current levels of 0 to 24mA with high precision. The device can operate on the current loop voltage supply from 10.8V to 40V. An SPI interface is used for communications. Since it is possible to connect two sensors to a single host controller, two 4-20mA loop drivers are required. Those can be connected using a daisy-chain method.

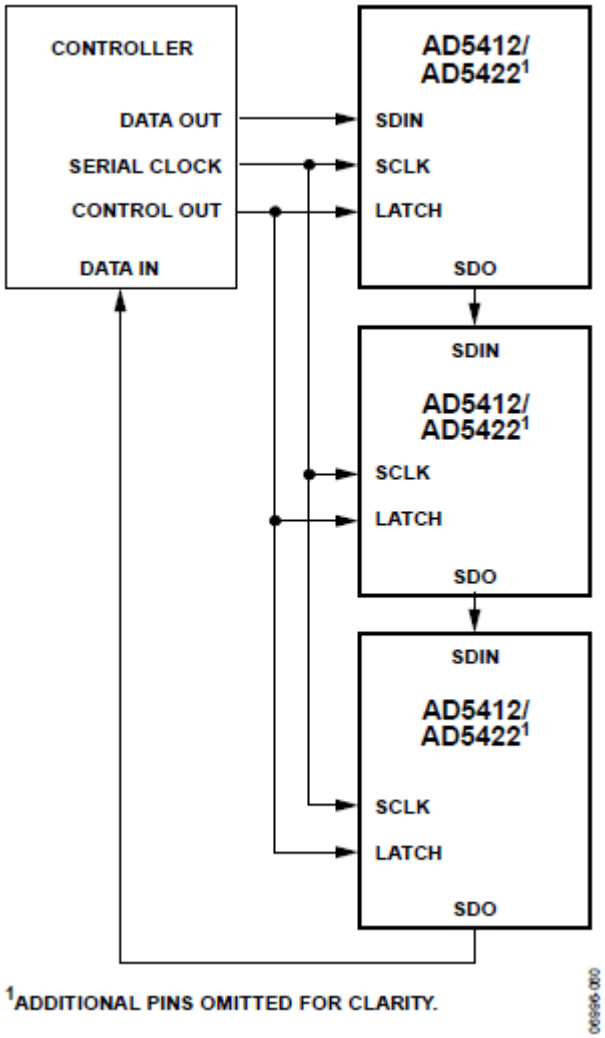


Figure 18: Daisy-chaining AD5412 [27]

3.4.4 Two-state relay

To indicate presence of the alarm, it is sometimes required to provide simple interface that would open or close a circuit. This can be achieved by using a solid-state relay. Since there are two sensors that can be connected to a single host, two relay outputs are necessary.

3.4.5 USB

A USB is a modern interface standard, which was released in 1996. USB has replaced COM port on most of the modern personal computers, and is currently the most widespread interface for connecting devices to a PC.

USB can provide both power supply and data communication. According to the standard, a USB port up to 2.0 supports a single +5V power wire, a ground wire, and two differential twisted-pair data lines (D+ and D-). A USB 3.0 standard features additional wires, however, being backwards compatible with USB 2.0. Since the features of USB 3.0 (such as an increased speed) are not necessary in an oil sensor host controller, they will not be implemented.

USB data rates depend on the generation. USB 2.0 allows for baudrates up to 60MB/s, which is more than sufficient for all the oil sensor needs.

3.4.6 Galvanic isolation

Some interfaces used in the host controller support long transmission lines, hundreds of meters of wires. In such conditions, formation of ESD is likely, for example, due to lightning strikes.

In such cases short but high-amplitude pulses can reach sensitive parts, such as controller, damaging them. To avoid it, interface drivers should be galvanically isolated from the rest of the circuit. Galvanic isolation is usually achieved by using optocouplers or electromagnetic couplers.

Galvanic isolation can also be used when the devices do not share common ground, or when interference should be minimized.

3.5 Design considerations

Aside from selecting the components, another important aspect is to make sure they can operate as a single system. This means, interfaces, power distribution and general structure should be decided upon.

3.5.1 Interfaces and ports

RCM6700 features a large number of ports and interfaces. However, the specifics of a host controller are such that a large number of end-devices should be connected to it. This makes distributing ports and pins a complicated task.

The following interfaces should be available from the host controller:

- USB
- SIG60
- Xbee
- RS232
- RS485
- Analog Output (2 channels)
- Signal relays (2 channels)
- I^2C (optional)

Except for the analog output, that uses SPI to communicate, and signal relays that are two-state and therefore only require a GPIO pin for every of them, other interfaces in their current implementation use a dedicated UART port.

Rabbit 6000 supports 6 serial ports (A to F) which can be used as both synchronous and asynchronous (UART). Four of the ports, A to D, can be used as SPI ports. There is a dedicated I^2C port, designated as Serial port G.

Some of this ports share pinouts. There are several options given for each port to use specific pins of the GPIO, which makes it possible to distribute pins so that all six ports can be used simultaneously.

Some ports have reserved purposes. For instance, Serial port A can be used for programming. Therefore, to support device programming mode from USB, which makes maintenance and development easier, this port should be reserved for USB interface.

To save time and reduce costs, some of development and debugging can be done before the actual boards were ordered, using an RCM6700 development kit as a prototyping system. To make it possible, ports should be chosen so that compatibility with the development kit is preserved. The Serial communications board of the development kit uses ports C and D for RS-232, and port F for RS-485. Therefore it is logical to reserve ports D and F for the same purposes.

Of the remaining ports, which are B, C and E, at least one should be dedicated as an SPI port. Only ports B and C can be used as SPI. As a synchronous interface, SPI uses, aside of transmitter (Master Out – Slave In, or MOSI) and receiver (Master In – Slave Out, or MISO), a dedicated clock signal (SCLK). Only certain pins for every port can be used. It was decided to use the serial port B as an SPI port.

The remaining ports, which are port C and port E, will be used for Xbee and SIG60 correspondingly, as UART ports.

The Serial Port G will be used for I^2C interface. While I^2C is not used in any of the applications currently, it might be reserved for further use or for additional connectivity.

There are additional requirements about general purpose pins for most of the interfaces. SPI devices require either Chip Select (CS) pin (which is the case for microSD card) or a Latch signal for analog output. Xbee and RS-232 support working in handshake mode, which makes it reasonable to add Ready-To-Send (RTS) and Clear-To-Send (CTS) signals for both. RS-485 requires a pin for transmission enable (DE) and a pin to switch terminal resistor on and off (TERM). There should be at least a pin for every relay used, some pins dedicated for analog output and SIG60, and an interrupt pin for I^2C.

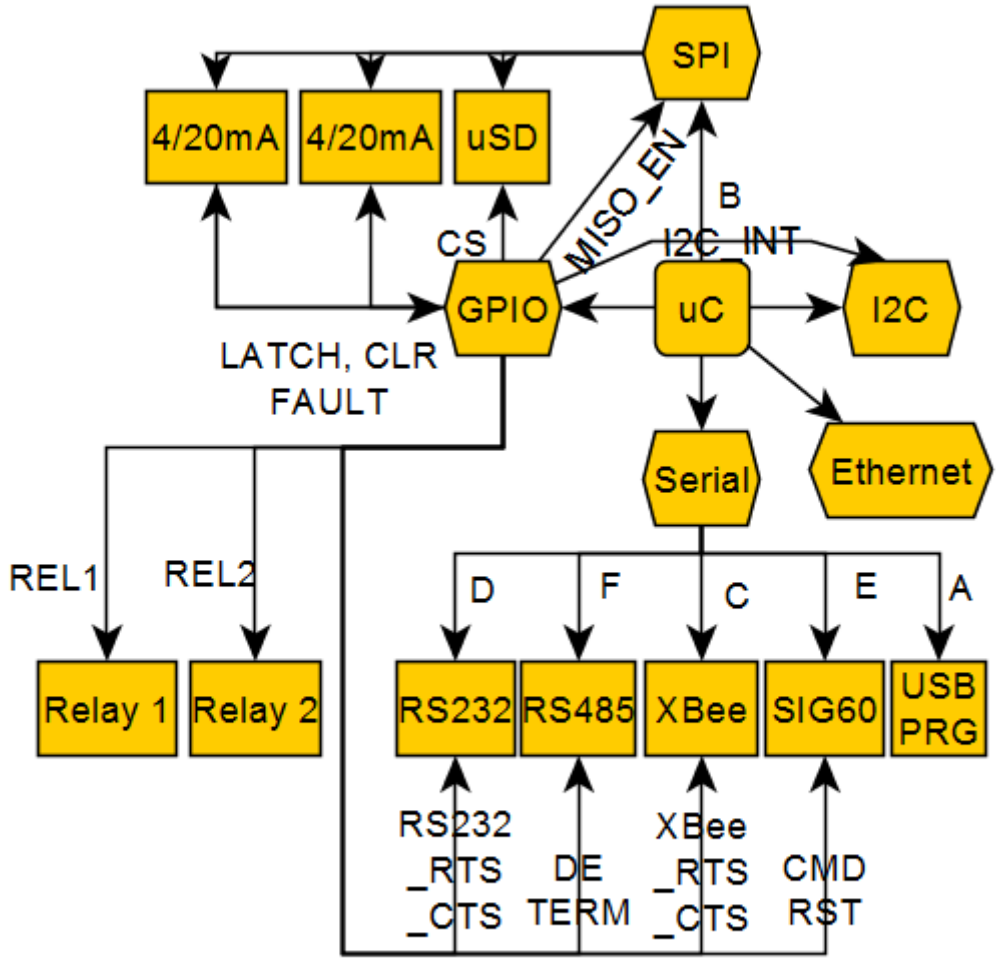


Figure 19: Block scheme of interfaces and pin requirements

3.5.2 Pinout

Some things should be considered before distributing pins available. RCM6700 module has a 52-pin PCI socket, not all pins however can be used as GPIO. Some pins are reserved for internal use, connectivity, power, et cetera.

GPIO pins can be grouped by parallel ports. There are 8 bytewide parallel ports on Rabbit 6000, Parallel port A to H (PA..PH). Each of the ports has 8 pins, totalling at 64 pins. Only ports A..E, however, are routed to the RCM6700 socket, and not all of the pins at every port are available, limiting the number of available GPIO pins.

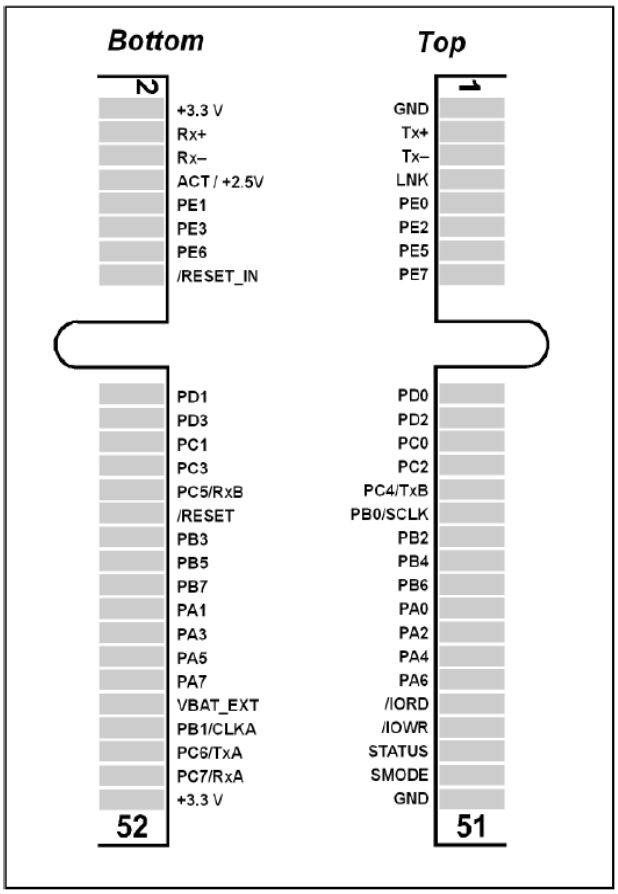


Figure 20: RCM6700 PCI Express header pinout [7]

First pins to be distributed are serial port pins. They have certain use restrictions, meaning that only selected pins can be used for a corresponding serial port.

Table 5: Serial ports A..F pin usage for RCM6700. [6]

Function	Serial Port A	Serial Port B	Serial Port C	Serial Port D	Serial Port E	Serial Port F
Transmit	PC7, PC6	PC5, PC4	PC3, PC2	PC1, PC0	PC6, PE6	PC2, PD2, PE2
Receive	PC7, PE7	PE5, PE4	PC3, PD3, PE3	PC1,PD1, PE1	PC7, PE7	PC3, PD3, PE3
Clock	PB1	PB0	PD2, PE2	PD0, PE0	PC5, PE5	PC1, PD1, PE1

Serial Port G (I^2C) uses pins PE0 and PE4 for Data (SDA) and pins PE1 and PE5 for Clock (SCL).

To ensure that all the ports can be used simultaneously, the following distribution scheme was developed.

Table 6: Pin distribution by serial ports

Port	Used by	Tx	Rx	CLK
Serial Port A	USB	PC6	PC7	PB1
Serial Port B	SPI	PC4	PC5	PB0
Serial Port C	Xbee	PC2	PC3	
Serial Port D	RS-232	PC0	PC1	
Serial Port E	SIG60	PE6	PE7	
Serial Port F	RS-485	PD2	PD3	

For I^2C, pin PE0 was selected as Data pin (SDA) and PE1 as Clock pin (SCL)

As mentioned above, there are some additional pins that should be used as GPIO. They are described below.

- SPI
    - MISO_EN (MISO Enable) – to additionally separate MISO signal from the internal SPI whenever external SPI is not used. Internal SPI is used for memory access within the module and utilises the same pins as external SPI.
    - CS (MicroSD card Chip Select) – to give command to the SD card to receive SPI data
    - LATCH (Analog Output Latch) – essentially a Chip Select, for daisy-chaining among analog output ICs
- Analog Output
    - CLR (Clear) – used to reset the analog output
    - FAULT – signal produced by analog output when it encounters a problem (usually to indicate an open-loop)

- SIG60
    - CMD (Command) – A signal for the SIG60 IC to enter command mode
    - RST (Reset) – A signal to reset SIG60 IC
- RS-485
    - DE (Data Enable) – A signal that indicates that the host controller is to start transmission on RS-485 bus
    - TERM (Terminal) – A signal to switch the RS-485 terminal resistor on or off
- RS-232, Xbee
    - RTS, CTS – Handshake signals, to indicate the state of readiness of the modem and the controller to receive
- Alarm indication relay
    - REL1, REL2 – a pin for every relay.
- I^2C
    - INT (Interrupt) – An interrupt indicating that there is a data for the I^2C bus to arrive

In total, 16 GPIO pins are necessary. The pins should not be in conflict with the ones used by serial ports, and they should not be placed on reserved spaces. While some of functional pins of the RCM6700 can be used as GPIO during normal operation (such as status pins), this might cause unnecessary conflicts and should be avoided when possible.

There is a moderate number of vacant GPIO pins left on the RCM6700 aside from those used by serial ports. Therefore, the pins can be distributed with a little reserve.

Following distribution was made, housing all the necessary pins.

Table 7: RCM6700 GPIO pin distribution

Signal	Used by	Direction	Pin
MISO_EN	SPI	Out	PB2
CS	MicroSD	Out	PB3
LATCH	Analog Output	Out	PA4
CLR	Analog Output	Out	PA6
FAULT	Analog Output	In	PB7
DE	RS-485	Out	PD1
TERM	RS-485	Out	PD0
CMD	SIG60	Out	PA1
RESET	SIG60	Out	PA0
REL1	Relay	Out	PA3
REL2	Relay	Out	PA2
RS232_CTS	RS-232	In	PB5
RS232_RTS	RS-232	Out	PA5
Xbee_CTS	Xbee	In	PB6
Xbee_RTS	Xbee	Out	PA7
I2C_INT	I^2C	In	PB4
	Reserve		PE2
	Reserve		PE3
	Reserve		PE5

As can be seen from the table, there are still 3 vacant pins available. Those can be used for further extension of the functionality.

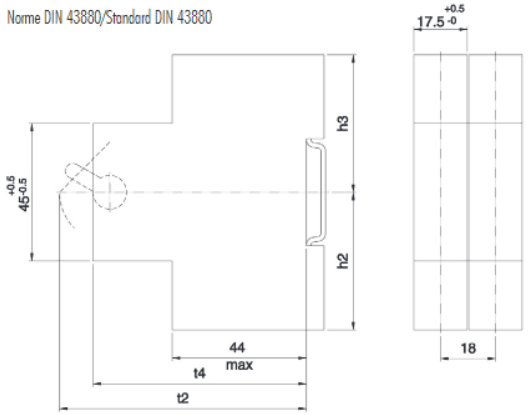
3.5.3 Device enclosure and layout

The host controller is intended for operation in an industrial environment, therefore its layout and body design have certain peculiarities. Dimensions should possibly be compact, but since the device is intended for stationary use, they do not bear critical importance. Water-tightness or mechanical toughness of the device are not required in all cases, since the controller is intended to be placed in enclosed space.

In modern industrial practice, automation controllers and other equipment is often installed in equipment cabinets. Equipment is usually placed on special fixtures called

DIN Rail. Those are special rails, that are located on a wall of a cabinet or a stand, to which equipment is fixed. DIN rail makes it easier to install equipment and save space by installing equipment on a wall, granting easy access from the front side.

Taking these considerations into account, it was chosen to select a DIN-rail mountable enclosure. The dimensions for such enclosures are regulated by DIN Standard 43880.



DIMENSIONI/DIMENSIONS	1	2	3
H2 max	45	55	75
H3 max	45	55	75
Ampere	Max 25 A	Max 63 A	Max 100 A
T2 max	55	70	92,5
T4 min	52	52	52
T4max	T2 max	T2 max	T2 max

Figure 21: Enclosure dimensions according to DIN 43880 [28]

Another consideration is the PCB layout. There were two main alternatives: a multi-PCB layout, stacked one above another, or a single board “flat” layout.

Stacked PCBs can make the design more compact, allowing for smaller enclosure size and more efficient usage of space. It makes the construction more modular, allowing for easier replacement of components when necessary. However, PCB assembly requires higher precision, which is not always easy to guarantee. Besides, internal components are less accessible. Placing interface connectors becomes more complicated. Therefore it was decided to choose a flat single-board design.

Taking all that into account, a modular enclosure produced by Italtronic was selected, Modulbox XTS 12MXTS B. [28] It has standard dimensions in accordance with DIN 43880, a terminal slot for connectivity, and a DIN rail clamp. Its shape is adequate for a single-PCB layout.

3.5.4 Voltage and power requirements

All the components on the board can be grouped by their voltage requirements

- 3.3V – RCM6700, Xbee, SIG60, microSD card reader, RS-232 driver
- 5V – RS-485 transceiver, USB driver, relays
- 12V – Analog output

To simplify the design, it was decided to create all these voltages directly from the input power supply using dedicated DC/DC converters.

The following calculations were done to find the power requirements. Only the most demanding components were taken into account. Data was taken from component datasheets, and the worst case scenarios were assumed.

Table 8: Power requirements for host controller components

Component	Voltage	Max. current	Max. Power	Source
RCM6700	3.3V	260 mA	0.86 W	[7]
XBee	3.3V	55 mA	0,18 W	[19]
SIG60	3.3V	50 mA	0,17 W	[15]
Micro SD	3.3V	100 mA	0,33 W	Various datasheets
Total for 3.3V		465 mA	1,53 W	
RS-485	5V	250mA	1,25 W	SP3032 datasheet
USB	5V	Powered by bus, not necessary to take into account		
Analog Output	12V	25mA * 4 = 100	1,2 W	[27]
Total power consumption:			3,98 W	

As can be seen, total power consumption will be around 3,93 W, with every tier of consumers (grouped by required voltages) requiring from 1.15 to 1.53 W.

For given purpose, and taking into account a necessary reserve of power, it was decided to use Recom RS-series regulated DC-DC converters with galvanic isolation. The series is rated for 2 W and features a nomenclature of various input and output voltages, including those necessary.

3.6 Circuit schematics design

For the circuit composition and further PCB design, Altium Circuit Studio software will be used.

3.6.1 Project structure

In Altium, it is possible to separate different parts of a project (a single printed circuit board) into dedicated documents called sheets. Altium offers two approaches to organize a project: a flat design (all sheets are equal and can be regarded as one schematic cut into smaller parts) and a hierarchical design (sheets vary in their role, as lower level sheets are used to fulfil higher level sheets).

A hierarchical structure allows for better representation of multiple structural blocks of the project, and offers such possibilities as reusage of some blocks, making it easier to use same complicated elements of schematics twice or more. An example in the curent project is the analog output, which is to be used twice. For these grounds, hierarchical structure will be used in this project.

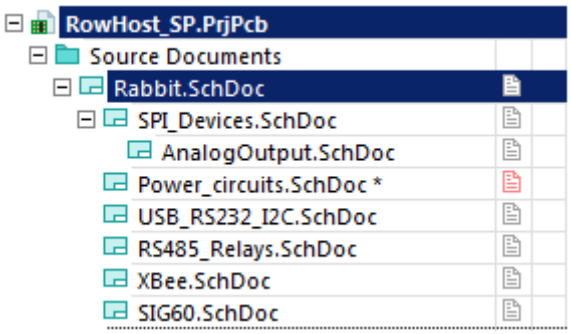


Figure 22: Host controller project structure

3.6.2 Microcontroller sheet

The main document of the project is Rabbit.SchDoc, which houses the PCI Express header used to connect RCM6700 module to the board, and other documents as structural blocks. Those blocks are represented on the current document by a special component called Sheet Symbol. Connectivity between those blocks and the document is implemented by input and output ports on the corresponding sheets, which are shown as Sheet Entries on the main document.

The microcontroller header, mini PCI Express, is a 52-pin header, with footprint and symbol taken from Altium Component Vault. To provide clear and unobstructed view, as well as improve readability, Net Labels are used in the schematic instead of wires.

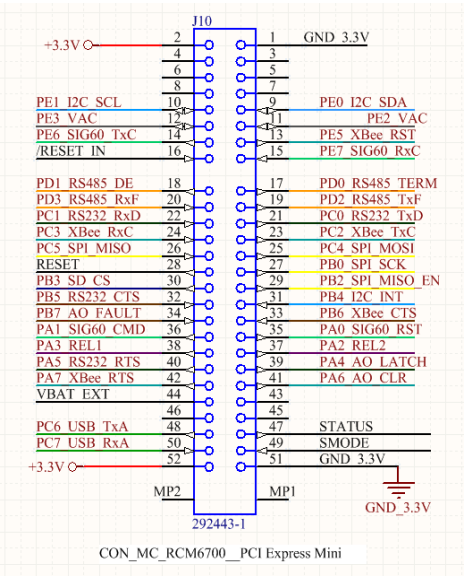


Figure 23: Mini PCI Express header

Other blocks present on the board are component blocks. These represent other sheets, and with an exception of power circuit block, have inputs and outputs to exchange signals with the controller.

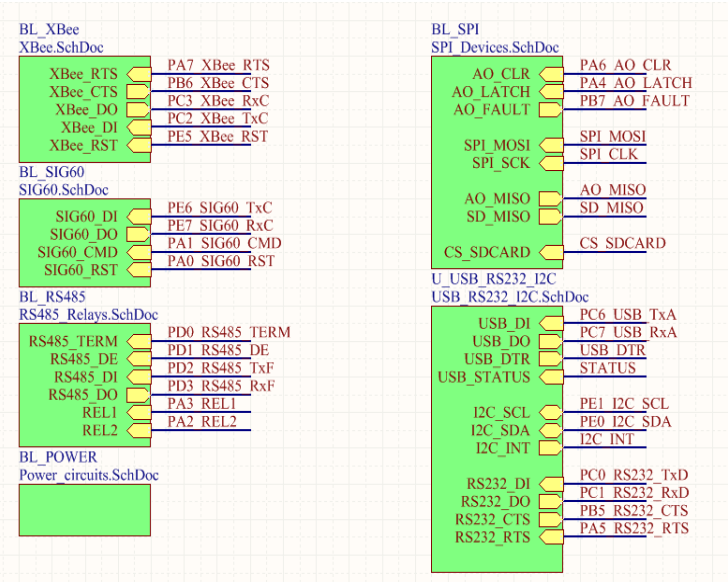


Figure 24: Peripheral blocks

To ensure proper signals on the SPI bus, the SPI signals are routed through the bus drivers, which are also on the same sheet. Some other signals are routed through a driver as well.

3.6.3 Power circuits

Taking into account the power requirements listed above, it was decided to use Recom RS series 2W converters.

The device is intended for 9..36VDC. Output voltages required are +3.3 V, +5 V and +12 V. Converters should be selected from the datasheet. [29]

Table 9: Selected DC-DC converters

Part Number	Input Range	Output Voltage	Output Current
RS-243.3SZ	9-36 V	3.3 V	500 mA
RS-2405SZ		5 V	400 mA
RS-2412SZ		12 V	166 mA

All three converters share the same input. First there is a standard power socket, which takes 9..36VDC from an external power source, such as 220V AC/DC adapter. For additional safety, a PolySwitch resettable fuse by Littelfuse is installed.

To ensure that voltage polarity is observed by user, and to avoid damage if polarity is reversed, a Schottky diode is installed on the line. To provide some protection from overvoltage, a Zener diode is added in parallel. Several capacitors are also added to remove any voltage spikes that may occur in the power line.

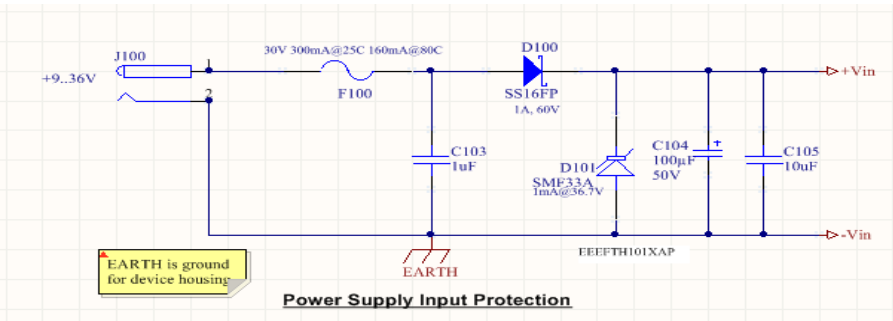


Figure 25: Power supply input for the converters

Three power converters accept input voltage from Vin and convert it to the required voltages. These converters offer galvanic isolation for extra safety. Grounds are connected through a capacitor and a resistor to avoid the difference of potentials between isolated grounds. For additional stability, some capacitors are added at the output of each power converter.

3.6.4 SPI Devices

There are two devices in the host controller that are intended to use SPI: a micro SD card and the analog output.

RCM6700 uses SPI to access internal memory, therefore it was decided to route the SPI signals through a bus driver to ensure the internal high-speed SPI is not affected by parasitic capacitances and other stray effects of the external circuitry. MISO signal can additionally be disconnected by use of a logic buffer to ensure no interference or false signals.

Both SD card and analog outputs share the same clock, MISO and MOSI signals. SD card has its own Chip Select signal (CS), while analog outputs are daisy-chained. Analog outputs also share the command signals, such as Latch (a command to the analog output chip to read the data in the SPI buffer) and Clear (to reset the chip).

Since there are two analog output chips to be controlled by a single Latch command, some way to send individual commands to each chip is required. The method used is called daisy-chain. A MISO (output) signal of one chip is connected to MOSI signal of another, and Latch signal is shared.

Analog output signal lines may stretch for hundreds of meters. In such conditions, formation of ESDs is likely on the wires. To ensure that a discharge does not damage the controller and other circuitry, and to prevent damage to inner circuitry occurring due to incorrect connection of outputs (such cases are not unheard of), it was decided to route the analog output signal lines through a dedicated galvanic isolation chip.

The IC selected was ADuM1400 from Analog Devices. It has 4 channels, supports level conversion and can withstand ESDs of high voltage (common mode transient immunity up to 25kV/us). [30] Two of these ICs are required to cover all the signals.

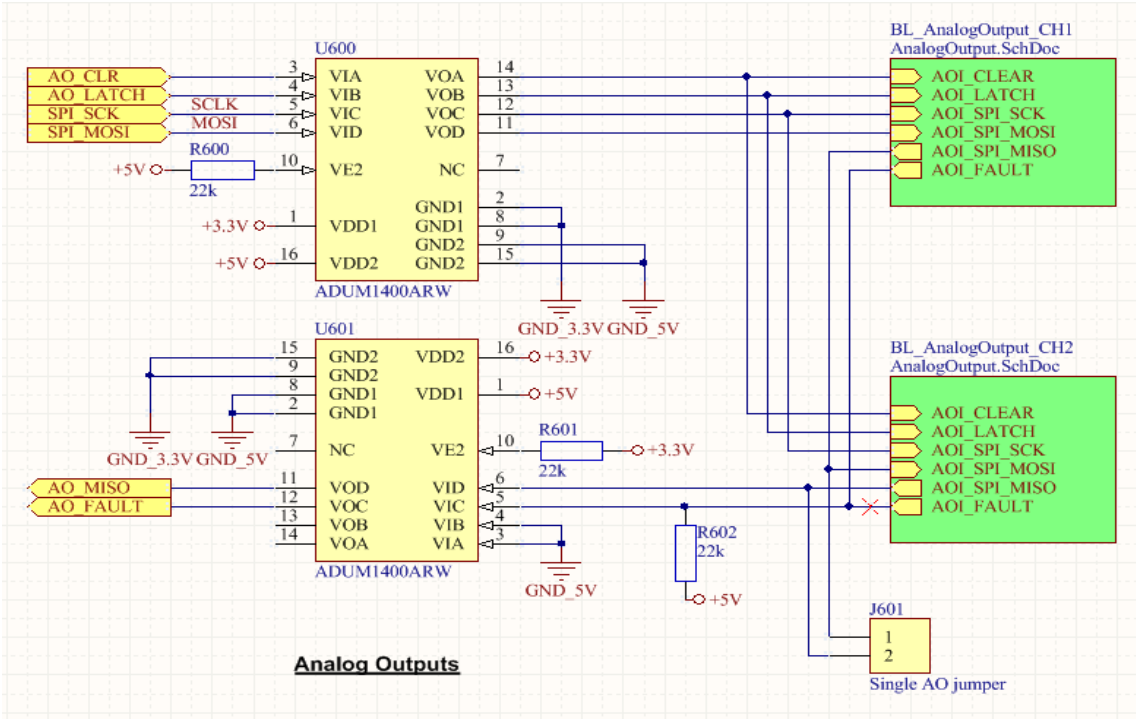


Figure 26: Daisy-chained analog output blocks with galvanic isolation

3.6.5 SIG60 and XBee

There is just a small number of signals that is required for Xbee. Since Xbee SX module series is supposed to work “out of box”, almost no circuit design is involved.

There are 5 signals to be used: Tx, Rx, CTS and RTS (Clear-to-send and Ready-to-send, handshake commands), and the Reset signal for power-on initialization of Xbee module, or to re-establish connection in case of interruptions.

SIG60 requires more complex circuitry to operate. There are four signals required: Reset, Data Out, Data In and Command mode. Aside of the usual components, it requires an external quartz oscillator and a ceramic filter. It is designed to operate with a 4 MHz quartz oscillator which should be connected to corresponding pins of the IC. Oscillator should be grounded with two capacitors.

Ceramic filter should be selected according to the datasheet, and tied to corresponding pin using a capacitor.

Analog supply (AVdd) should be connected to 3.3V power supply using a low-pass filter to avoid interference.

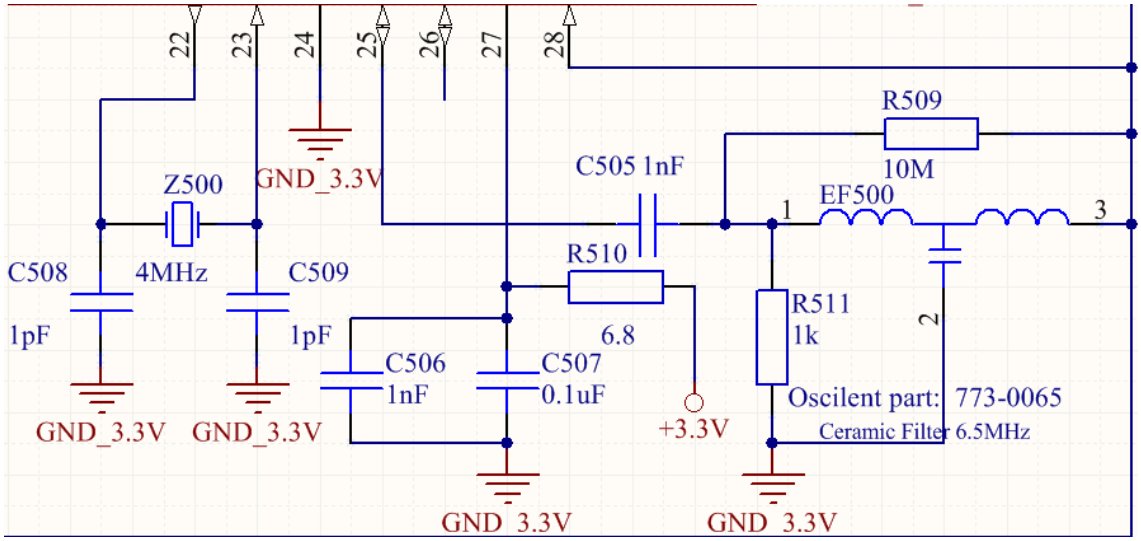


Figure 27: Oscillator, ceramic filter and AVdd connection on SIG60

Receiver and Transmitter pins of SIG60 are connected to the sensor power line, and the DC component is removed using capacitors. The sensor power supply is connected to the main power input with inductors for additional stability.

3.6.6 RS-485 and Alarm indication relays

RS-485 is implemented on base of SP3082 driver by Exar. This driver operates on 5V voltage level.

There were various drivers available for both 3.3V and 5V logic. With 3.3V, voltage conversion and 5V supply would not be necessary, which would have simplified the design. However, based on some research, it was decided to use 5V logic with RS-485. The reason for this is that when operating on 3.3V, a driver requires higher sensitivity, which results in higher price, lower reliability and capabilities.

RS-485 circuitry is moderately simple. It receives 3 signals, DI (Data In), DO (Data Out) and DE (Data Enable), and produces differential signal on lines A and B. Bias resistors are installed on the output lines (on inverted line – 4.7k pull-down resistor, on non-inverted – 4.7k pull-up resistor) to ensure the line is always in a known state. Transient voltage suppressor diodes (SM712 and SM05) are installed for overvoltage protection.

In some use cases, it is required to install a terminal resistor of 120 Ohm on the line. For user's convenience, it was decided to make the resistor software-switchable, so that it can be enabled or disabled by giving a corresponding command to the host controller. This was implemented by using a solid-state relay AQY214S, which is connected in line with the 120 Ohm terminal resistor.

Recommended current for the activation of LED in the relay is 5 mA. To ensure stable and safe operation, it was decided to use a transistor to control the relay. A well-suited transistor for that role is PDTC124ET resistor-equipped transistor by Nexperia.

According to the datasheet [32] , the dropout voltage is between 1.0 V and 1.2 V. Therefore, to limit the current, a 510 Ohm resistor was added.

As the range of RS-485 lines can be quite long, and formation of ESDs and overvoltages due to lightning strikes is likely, to protect inner circuitry and microcontroller it was decided to add galvanic isolation. An ADuM1401 galvanic isolation was used. It is the same series as aforementioned ADuM1400, but with one channel direction reversed. This way, a single IC is enough to accommodate all three RS-485 signals and the terminal resistor relay control signal.

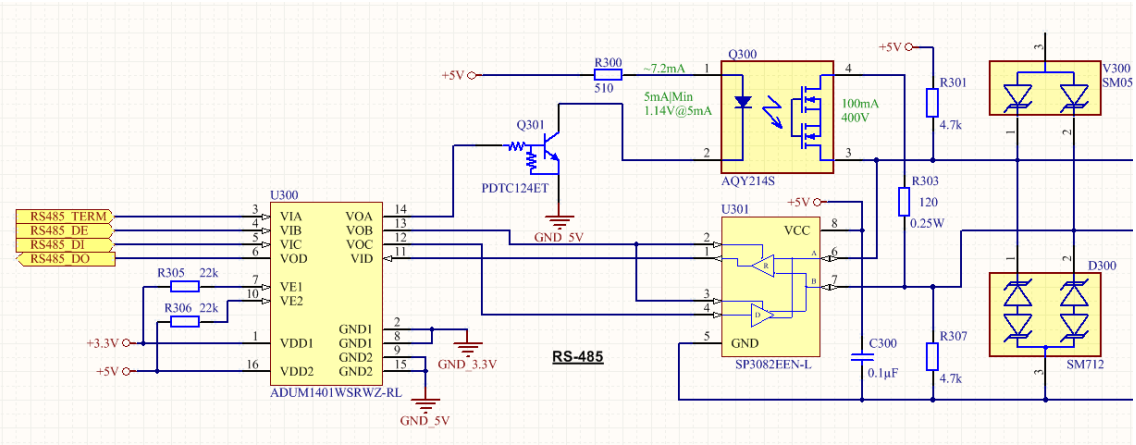


Figure 28: RS-485 circuitry with terminal resistor relay and galvanic isolation

In addition to the RS-485 interface, there are two relays to indicate presence of an alarm. These are the same AQY214S relays as the ones used for terminal resistor switching. Since the relays are galvanically isolated themselves, additional isolation is not necessary. Instead, the signal will come directly from the microcontroller. Similar to

the terminal resistor switching relay, there will be a PDTC124ET resistor-equipped transistor. RCM6700 GPIO provides +3.3V signal, as opposed to ADuM1401 which was set to produce +5V on its output side, therefore the current limiting resistor will have a smaller value, 270 Ohm.

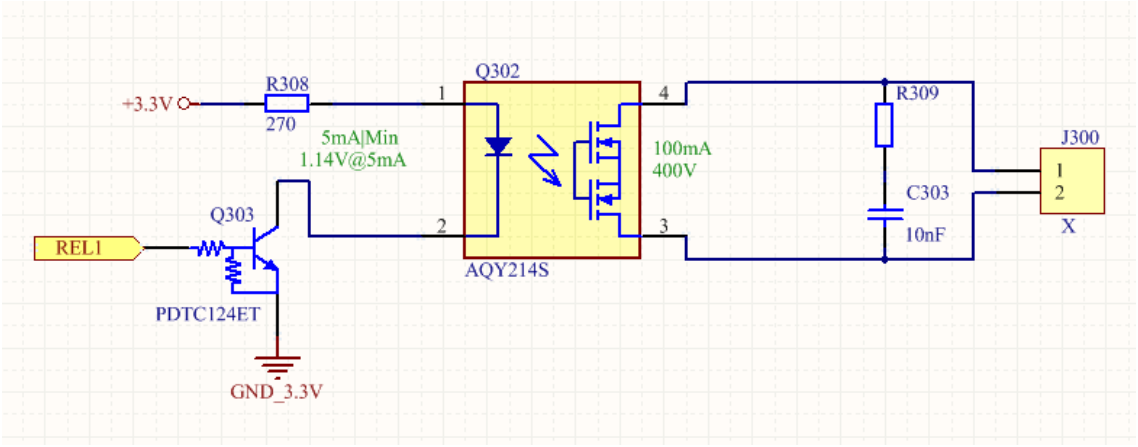


Figure 29: Alarm indication relay

3.6.7 Digital communications

Aside of the aforementioned interfaces, the remaining ones are USB, RS-232, I^2C and Ethernet.

Since RCM6750/RCM6760 modules come pre-equipped with Ethernet socket, and routing of external socket would require excessive precision and quality, it was decided not to place an Ethernet socket on the board, but use the one installed on the module.

USB interface will be implemented using an FT232RL USB driver by Future Technology Devices International. The IC will be powered by the bus itself.

The two signals, Data In and Data Out, are connected to the microcontroller directly. The programming mode also makes use of DTR and Status signals. The output is connected to the USB type B socket. The Vcc pin is used to power the FT232 IC. The differential data lines (D+ and D-) are connected to corresponding pins on FT232, and ground is connected to the +3.3V ground plane. A transient voltage suppressor is installed between D+ and D- to provide additional security. Shield is grounded through the conductor.

I²C interface is implemented by the microcontroller, and is routed to a socket using an LTC4311 I²C bus accelerator by Linear Technologies. RS-232 is implemented using an AD3202 line driver by Analog Devices.

3.6.8 Programming mode support

RCM6700 module can be programmed directly from the Serial port A. To expedite production and testing, it was decided to add the support of on-board programming of the module using the USB.

For the controller to enter programming mode, SMODE pin should be pulled to high level. Besides, for additional synchronization, DTR pin is connected to RESET_IN pin on the controller. Both of those connections should be disengaged in normal operation.

To provide that, a jumper will be installed on the board. An 74HC125MX buffer will be used to connect SMODE to +3.3V, and RESET_IN to DTR, when the jumper is populated. The same buffer is used to disconnect MISO signal from the microcontroller when SPI is not in use.

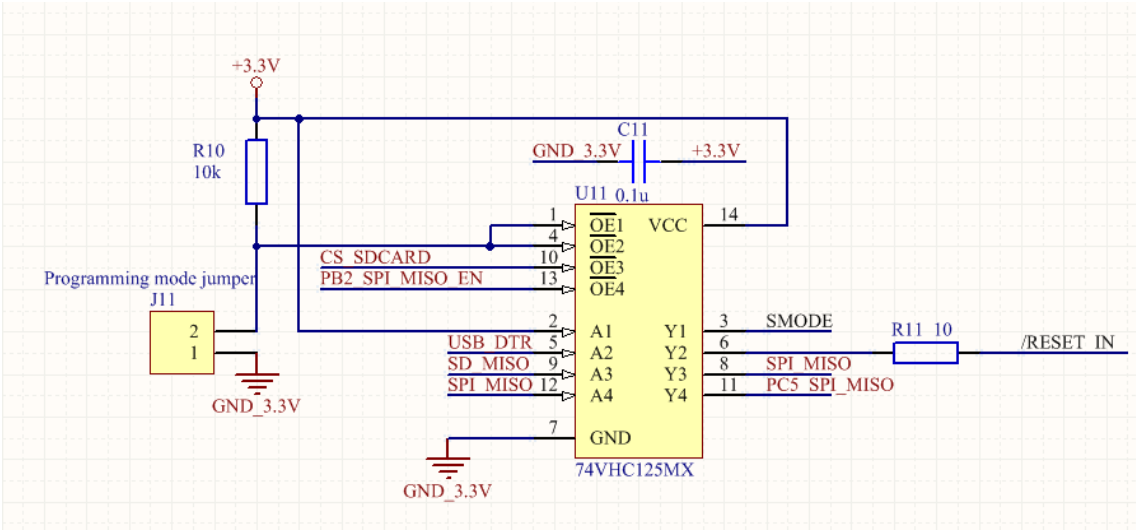


Figure 30: Programming mode jumper and 74VHC125 buffer

4 Software

The software part is dedicated to design process of firmware for the host controller.

4.1 Dynamic C

Rabbit 6000 uses its own proprietary dialect of C language, which is called Dynamic C. It was developed by Rabbit Semiconductors for Rabbit series microprocessors applications.

4.1.1 Dynamic C 10.72 IDE

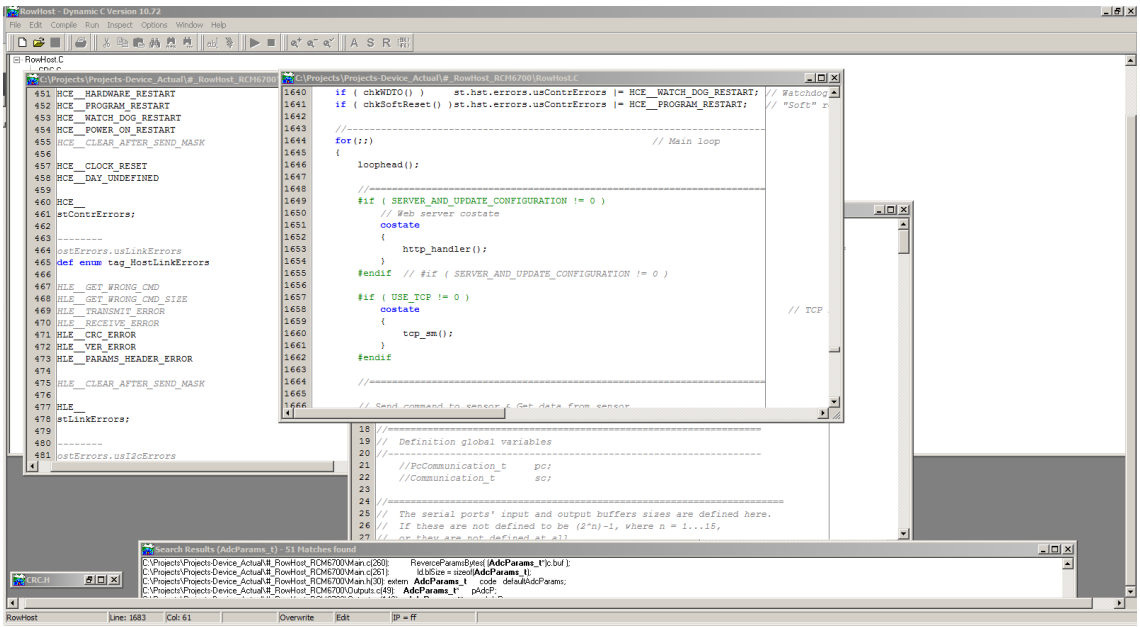


Figure 31: Dynamic C 10.72 IDE

There is a free IDE available for Dynamic C. The latest version by March 2018 is Dynamic C 10.72. This IDE integrates such tasks as editing, compiling, linking, loading and debugging into one software solution. [21]

For editing it includes a built-in text editor. Although it may be less powerful than text editors in more prominent software, such as Microsoft Visual Studio, or dedicated code

editors such as Notepad++, it still provides sufficient functionality for editing code without need for third-party software. It supports creating projects with their own custom settings. Such functionality as search within project, adding files to project, defining macros either in code or in special menu option, et cetera, is supported.

A compiler and linker for Dynamic C is also included. Compilation and linking error and warning messages will be displayed, if any are present. Number of errors and warnings to be displayed can be customized.

There is also a support of a variety of compiler directives that provide such functionality as name definitions, conditional compilation, inclusion of libraries, insertion of assembly code, and other functions to expand the available toolkit.

By using conditional blocks of compiler directives, such as #if or #ifdef, it is possible to set initial values and enable or disable whole blocks of code by changing a single variable or a macro. This helps to simplify modifications, or compile firmware with required functionality for certain use case, while preserving the code that can be used in other applications of the same product. Since those blocks that are disabled by compiler commands are effectively ignored by compiler, they put no strain on the program if disabled.

A compiled program can either be saved as a binary file to be uploaded to a controller later or written directly to the memory of a controller from the IDE.

Rabbit 6000 and Dynamic C 10.72 IDE also support run-time debugging on a controller that is connected to the PC. There is such functionality available as placing breakpoints, step-by-step execution, watch expressions, stack tracing, et cetera. It is possible to display text console messages within the IDE by using corresponding commands in the code.

4.1.2 Differences between C and Dynamic C [21]

There is a number of enhancements offered by Dynamic C, that are not supported by standard C language. Those enhancements serve the purpose of simplifying the development of embedded systems and real-time applications.

One of the major improvements over standard C is introduction of cooperative multitasking. This is implemented by using costatements and cofunctions.

Since Dynamic C 10.54, remote firmware update has been introduced for some board types. There are some additions to the language that make update process easier to implement.

Function Chaining is a concept that allows special segments of code to be distributed in several functions. When a chain executes, all the segments of this chain execute. Function chain makes calling some complex tasks, like initialization, data recovery, et cetera, more efficient.

Data protection is supported by using *shared* and *protected* keywords.

There is a number of tools that allow use of extended memory (*xmem*) and let the programmer take control of memory management.

There is also a number of minor changes and requirements in Dynamic C that are not present in standard C.

4.1.3 Cooperative Multitasking in Dynamic C

A usual microcontroller program usually consists of a main infinite loop that makes sure that the program runs indefinitely, unless explicitly required to shut down. The program flow is usually sequential, meaning that the next task is only started after the previous one has finished working. If a function requires to wait or takes a long time to complete, it usually means that the execution is paused until the task is complete. Multi-tasking and parallel states in state machines can be difficult to implement on traditional architecture and using means of standard C language.

In Dynamic C, costatement is a block of code that supports cooperative multitasking. Costatements start to execute from the first one sequentially, like in most state machines. However, whenever there is a wait or yield, execution proceedes to the next costatement, while the current one is suspended, its state being saved. When all the other costatements either yield or complete, and the execution reaches the suspended costatement once again, instead of starting from the beginning of the block, it starts from the exact moment where it has been suspended, provided that the conditions of

resuming the process are met (a defined wait time has passed, a function returned a value, et cetera). Otherwise, execution is once again yielded to other costatements for another iteration.

There are three main methods of yielding execution to other costatements: *waitfor*, *yield* and *abort*. Waitfor expression takes an argument (a function or conditional expression) and checks if it is non-zero, in which case it proceeds to the next statement within the costatement. Otherwise it will yield execution to the next costatement. Yield passes execution to the next costatement when hit first time. Abort statement simply exits the costatement, which is next time started from the beginning, similarly to *break* in loops.

Cofunction is essentially a function that may yield when a wait is necessary, for instance, when waiting for data to arrive on a serial port. They are called using *waitfordone* (which can be abbreviated to *wfd*) statement. Like a normal function, cofunctions can accept parameters and return values. However, they cannot be assigned to function pointers.

4.1.4 Programming and debugging the microcontroller [21]

Rabbit 6000 together with Dynamic C 10.72 IDE support a convenient and sufficiently powerful debugger.

The controller may run in Programming mode or autonomous mode ("Run mode"). In autonomous mode, it does not require connection to a computer and may run independently, executing the program that is present in its memory. In this mode, debugging using the means of IDE is not possible, since the controller is not polled by the PC.

The Programming (or bootstrap) mode allows downloading the program onto the controller, and supports debugger operations. In this mode, instructions are fetched from the peripheral port and written into either memory or a certain register. Autonomous work is not possible in this mode.

The mode is chosen by setting SMODE pins to either high or low. With SMODE pins set low, Rabbit 6000 begins fetching instructions from the memory mapped into bank 0. Depending on specific SMODE pin, when one of them is set high, instructions will be

accepted either from Serial Port A, serial flash bootstrap port or slave port. In case of RCM6700, only one SMODE pin is available, limiting options to the port A. When using RCM6700 Development kit, a jumper on the interface board can be populated to enter programming mode, connecting SMODE to +3.3V and allowing the controller to be programmed and debugged using the USB. Otherwise, SMODE pin on the PCI header should be set at high level by the user.

There is support of various debugging tools, such as breakpoints, step-by-step execution, variable watches. Step-by-step debugging might be complicated by parallel processing.

4.2 Code composition

As for any complex project, the code requires a thorough thinking and a plan before implementation.

4.2.1 Functional requirements

The firmware for RCM6700 should be capable of doing the following:

- Poll the sensor autonomously
- Receive data from sensor, make an analysis and set alarm status if necessary
- Signal the alarm (or lack of) to the analog output and relays
- Listen to the interfaces that communicate with PC
- Accept commands from the PC and if necessary, pass them along to the sensor
- Report the current status (signals, state of alarm) back to the PC

The convenient multitasking mechanism that is available in Dynamic C can best be used for implementation of the task.

The host controller will poll the sensor from time to time, depending on the parameters defined. There is a number of commands that sensor can accept from the host controller:

- Controller commands

- Test link – a simple “ping” to test if the sensor is functioning and the link is intact
- Send signals – Send the collected data to the host controller
- Send type – Send the sensor type (to check if the sensor is supported)
- Restart – restart the sensor controllers

- Parameter commands
  - Set params – update sensor parameters with the one received from the host controller
  - Set default params
  - Request params – request parameters that are set at the sensor-host

- Firmware commands
  - Write – write the new firmware image to memory
  - Erase – erase the firmware image from memory
  - Program – replace the currently running program with the image from memory

There is a number of other debugging and technical commands.

The data received from the sensor will be analyzed by the host controller and the decision will be made whether to raise an alarm. If the alarm is raised, corresponding interfaces (relays, analog output) will receive a command to act accordingly to indicate an alarm.

The host controller will have to listen to the PC and if a command comes, it should be able to execute those commands. Those may include passing a parameter or a new firmware to the sensor, reporting current state or signals, changing the host controller parameters, etc.

4.2.2 Flowcharts

A flowchart can be used to demonstrate the processing logic of the firmware.

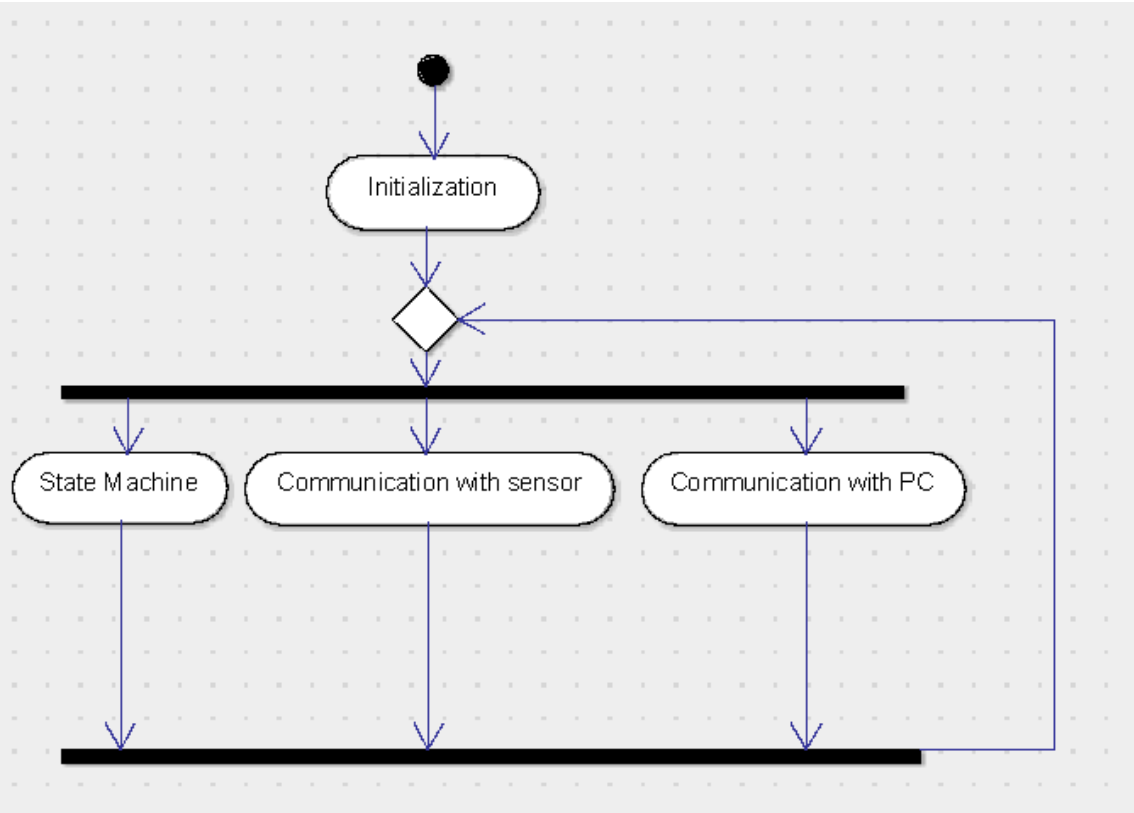


Figure 32: Firmware general flowchart

The first task is initialization. It assigns the values to all the global variables, reads parameters from non-volatile memory and initializes all the ports and interfaces. It also sets the first stage of the state machine.

After initialization, there are three costatement processes. They start one after another, but will yield whenever a task requires them to wait. Therefore they may be considered parallel. For instance, while one of the processes awaits a command from the PC, the other may poll the sensor.

The state machine is responsible for the initialization of sensors, by requesting their parameters and type, as well as periodic polling of sensors to request data. The period of polling can be defined by user.

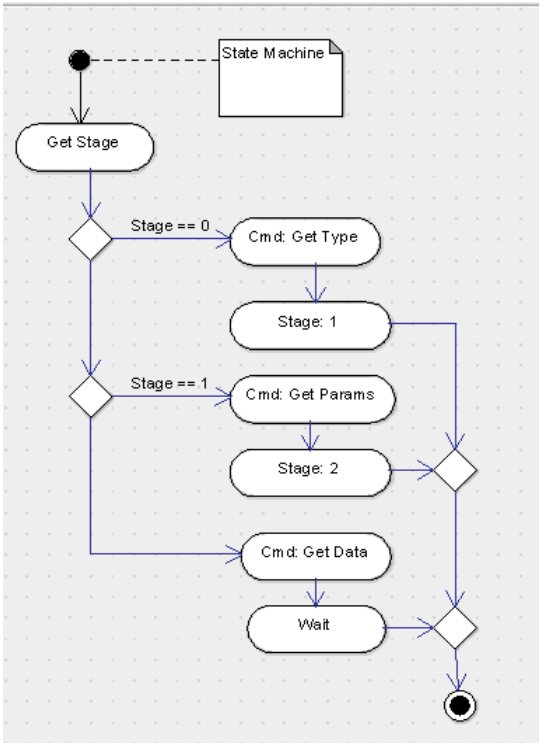


Figure 33: State machine flowchart

To initialize a sensor, first its type is requested. This way the host controller makes sure it is communicating with the right device. After the type, sensor parameters are requested and copied into memory of the host controller. On the final stage, the sensor is polled for data, which is then saved to memory and analyzed.

The state machine does not itself conduct communication, but rather sets a global variable to indicate the command required. This command will then be handled by the following co-process, which is responsible for communication with sensor.

The second co-process checks if there is any command pending. If there is none, it exits and the execution steps to the next co-process. If, however, there is some command for the sensor, be it a request for data or a new set of parameters, this command will be sent to the sensor. After the command has been sent, the co-process yields until a response has been received or the timeout has elapsed, in which case it will exit. If the response has been received, and if it has been confirmed valid (which means, both the addresses, command and the CRC are correct), it is then processed according to the initial command.

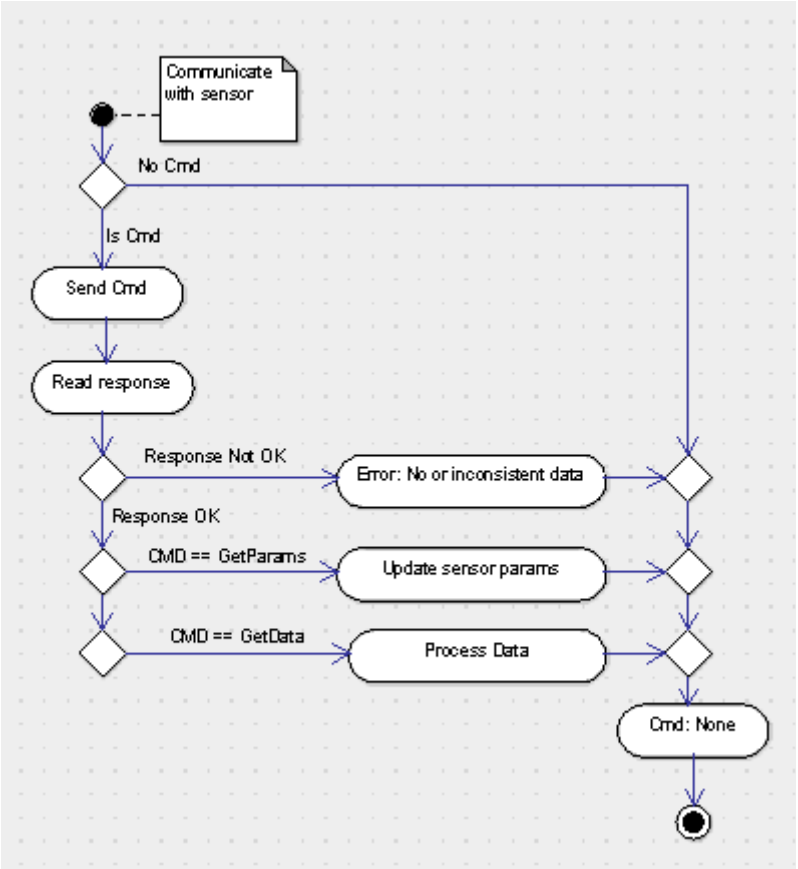


Figure 34: Communication with sensor flowchart

In case the command was to transmit the gathered data, the response will be saved and analyzed. The signals are saved to a buffer, and the analytical part will decide whether there is an oil spill or not. Once the decision has been made, the analog output and the relays will get the command to indicate an alarm. Depending on parameters, the state of alarm may be indicated differently. For instance, relays can be set to work as “Normally open” or “Normally closed”. Analog output can also provide different modes of operation. After that, the sensor state will be updated and the co-process will complete.

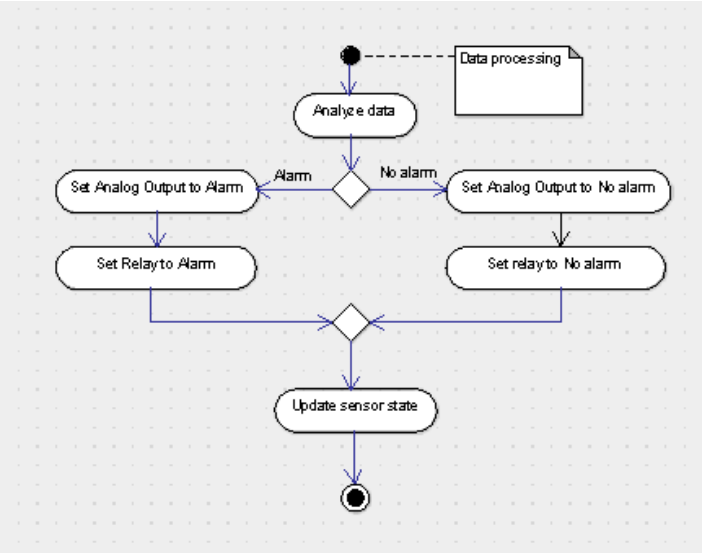


Figure 35: Data processing flowchart

The third and final co-process is responsible for communications with PC. It awaits for data on any of the available “external” interfaces (USB, RS-232, RS-485, Ethernet), and yields until that data arrives.

If the command has arrived and was found valid, it is implemented by the host controller. There is a number of commands that can be received from the PC. For some, only an action on the host controller side is required. For instance, if signal data is requested, it is taken from the buffer and passed along as a response. Other commands require interaction with the sensor. In this case, the sensor command is set into a corresponding global variable, and an acknowledge is sent to the PC. The co-process yields, and as the execution reaches the sensor communications costatement, the command to the sensor will be passed.

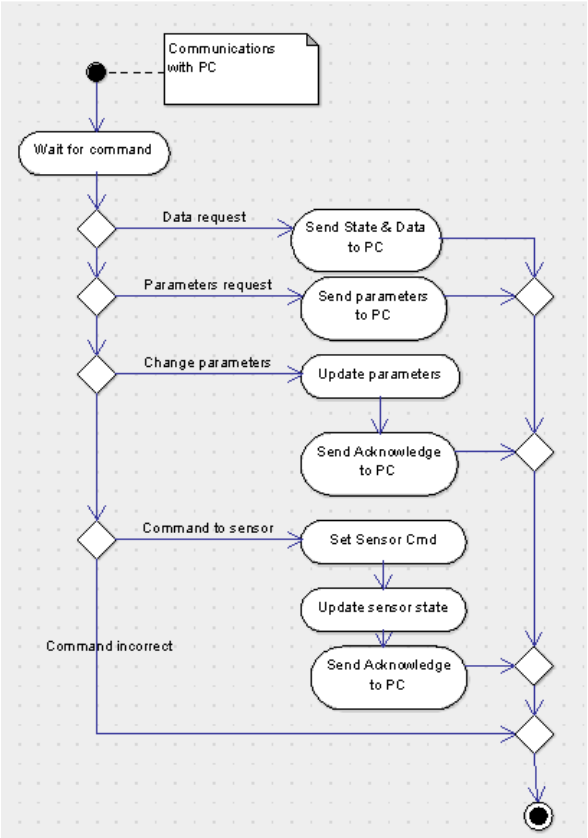


Figure 36: PC communications flowchart

5 Summary

The purpose of this thesis was to develop a host controller for an oil spill sensor. The host controller had to satisfy certain functional and technical requirements, including interface support and performance characteristics.

5.1 Overall progress

Although it was not possible to create a working prototype in time due to time constraints, the project has still accomplished its goals.

The design job was completed, with the schematics for all the components ready and the PCBs are in a state of development. The firmware code was written and debugged. Although there is not yet a fully-functional prototype available, partial functionality of hardware and firmware was tested on base of the RCM6700 Development kit. The host controller was able to communicate with sensor, create an alarm event and control the behaviour of its outputs properly.

The required technical parameters are expected to be met in the prototype. The power consumption will be around 6W, with voltage supply allowed from 9V to 36V, provided by the DC converters used. Support for the required interfaces and hardware was added in firmware and documented as the schematics. Radio link was tested to have range of up to 1km in good conditions with the intended antennas. The host controller successfully communicated with the sensor at 57600 bps.

The current state of the host controller allows for any necessary modifications and customization for specific user's needs, which was complicated, or in some cases impossible with the previous version of the product.

Due to the fact that most of the job, including communications, is done by the host controller, the sensor consumes less power and works more efficiently and reliably than the previous version. In addition, since the sensor is not accessible directly by user, an

effect of human factor on the reliability is decreased. Host controller, on the other hand, is more accessible for maintenance.

Taking all this into account, it can be said that the goals of the thesis have been fulfilled.

5.2 Future work

Currently, the PCBs are in a state of development. When they are manufactured, the fully-functional prototype should be constructed and tested. After that, the PC software should be written, or the existing adapted, to work with the new product.

As soon as everything is complete, the product can be shipped to the customers. The opinions of users will be crucial in the final evaluation of the product.

5.3 Possible improvements

Since the idea of the host controller was to simplify any modifications and adaptations, there is room for improvements of the current design. Further development of the product will be based on users' initial assessment and expectations.

Currently, the host controller only supports the wireless link between itself and the sensor. While this is adequate in most cases, a support for a radio link between the host controller and the PC would be a convenience worth considering.

The host controller uses a proprietary protocol to communicate with PC. In the past, some users have expressed a wish to have some widespread protocol available, to be able to get data directly from the host controller without need of a PC software. While this has not yet been implemented, such an option might be added if the future.

Some additional hardware might be required in certain use cases, such as an embedded GSM modem. This may be achieved in future without major difficulty.

As of now, the maximum number of sensors that can work with a single host controller, is limited by 2. While in most cases this is an adequate number, there might be implementations that would benefit of more sensors serviced by a single host controller. Such functionality might also be added in time.

References

[1] FLUCOmat FLU-103 prospect, Chemtronic GmbH, http://www.chemtronic-gmbh.de/images/chemtronic/PDFe/Prospect%20Flucomat_e_.pdf

[2] ODL-1610A Specification Sheet, TOA DKK, http://www.toadkk.co.jp/english/product/env/wq/i36ghb0000003xtm-att/ODL-1610A(E)_P2.pdf

[3] Fluorescence Fundamentals, TermoFisher, https://www.thermofisher.com/ee/en/home/references/molecular-probes-the-handbook/introduction-to-fluorescence-techniques.html

[4] Selecting a Microcontroller (MCU) for your IoT product, https://www.particle.io/mcu-vs-soc-vs-microprocessor-for-iot/

[5] RISC vs CISC, RISC Architecture, https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/

[6] Rabbit 6000 Microprocessor User's Manual, Digi International Inc., http://ftp1.digi.com/support/documentation/90001108_H.pdf

[7] MiniCore RCM5700/RCM6700 User's Manual, Digi International Inc., https://www.digi.com/resources/documentation/Digidocs/pdfs/90001191.pdf

[8] Raspberry Pi Hardware, Raspberry Pi Foundation, https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md

[9] RPi Hardware, https://elinux.org/RPi_Hardware

[10] ARM Processors, https://www.arm.com/products/processors

[11] Cortex-M Series, ARM, https://www.arm.com/products/processors/cortex-m

[12] Going back to the beginning of AVR with Vegard Wollan, Atmel, https://atmelcorporation.wordpress.com/2014/08/25/more-avr-history-with-vegard-wollan/

[13] Atmel AVR microcontrollers, Atmel/Microchip, http://ww1.microchip.com/downloads/en/DeviceDoc/AVROVERVIEW-02-11%20flyer%20-%20US%20-%20low%20res-4.pdf

[14] What is Power Line Communication, EE Times, https://www.eetimes.com/document.asp?doc_id=1279014

[15] SIG60 datasheet, Yamar Electronics Ltd, http://www.yamar.com/datasheet/DS-SIG60.pdf

[16] Frequently Asked Questions, International Telecommunication Union (ITU), https://www.itu.int/net/ITU-R/terrestrial/faq/index.html

[17] SRD Short Range Devices, Radio-Electronics, http://www.radio-electronics.com/info/wireless/srd/short-range-devices-cept-etsi.php

[18] Bluetooth radio interface, Radio-Electronics, http://www.radio-electronics.com/info/wireless/bluetooth/radio-interface-modulation.php

[19] Digi XBee SX 868, Digi International Inc., https://www.digi.com/products/xbee-rf-solutions/sub-1-ghz-modules/digi-xbee-sx-868

[20] Choise of frequency bands can really make a difference, Myk Dormer, Radiometrix, http://www.radiometrix.com/files/additional/choice-of-frequency-band-can-really-make-a-difference.pdf

[21] Dynamic C User's Manual, Digi International, http://ftp1.digi.com/support/documentation/019-0167_L.pdf

[22] Application Note 83: Fundamentals of RS-232 Serial Communications, Dallas Semiconductor, http://ecee.colorado.edu/~mcclurel/dan83.pdf

[23] RS485 serial information, Lammert Bies, https://www.lammertbies.nl/comm/info/RS-485.html

[24] RS-485 Tranciever Tutorial, Renesas Electronics, https://www.intersil.com/content/dam/Intersil/whitepapers/interface/rs-485-transceiver-tutorial.pdf

[25] Application Note AN-960: RS-485/RS-422 Circuit Implementation Guide by Hein Marais, Analog Devices, http://www.analog.com/media/en/technical-documentation/application-notes/AN-960.pdf

[26] Fundamentals, System Design, and Setup for the 4 to 20 mA Current Loop, National Instruments, http://www.ni.com/white-paper/6940/en/

[27] AD5412/5422, Analog Devices, http://www.analog.com/media/en/technical-documentation/data-sheets/AD5412_5422.pdf

[28] Italtronic Modulbox XTS 12MXTS B, https://docs-emea.rs-online.com/webdocs/139f/0900766b8139f866.pdf

[29] Recom RS series datasheet, https://www.recom-power.com/pdf/Econoline/RS-S_D_Z.pdf

[30] ADuM1400/1401/1402, Analog Devices, http://www.analog.com/media/en/technical-documentation/data-sheets/ADuM1400_1401_1402.pdf

[31] RS-485 compatibility between 3.3V-supply and 5V-supply devices, Clark Kinnaird, Texas Instruments, https://e2e.ti.com/support/interface/industrial_interface/f/142/t/93260

[32] AQY210S series datasheet, Panasonic, https://www3.panasonic.biz/ac/e_download/control/relay/photomos/catalog/semi_eng_gu1a_aqy21_s.pdf

All references to the web resources were last visited and found working on 2.05.2018.