Na prawach rękopisu

INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ
Raport serii PRE nr 48/2004

Design and construction of RoBall, a spherical, nonholonomic mobile robot

Marek Kabała
Marek Wnuk

Słowa kluczowe: mobile robot,
DC motor drive,
sensor,
control system,
microcontroller,
programming

Wrocław 2004

1 Introduction

The idea to build a laboratory spherical robot arised in the Unit of Foundations of Cybernetics and Robotics in connection with research works on kinematics singularities of nonholonomic robots [2]. The initial concept of construction of such a robot called **RoBall**, in particular - the concept of its driving system, was presented on the seminar on 4th May, 2000. The preliminary design of RoBall spherical robot has been described in [6]. The driving torques, provided by two DC motors, are related to 2DOF pendulum hanging in the centre of the spherical robot shell, what requires considering the robot's dynamics while modelling it's behaviour and designing control algorithms.

The preliminary simulations of such an object show [14] that taking into account all the real world constraints during the robot modelling is not an easy task. Experimenting with a real robot should help users avoid overlooking any essential phenomena. RoBall is intended to be a real object equipped with sensors, drives, and an embedded controller, what will enable users to implement and test their own control algorithms.

1.1 Former results

In order to test the basic concepts of the drives and the sensors, a two wheel mobile robot, called MK cart was constructed (figure 1). The results of research based on MK robot were published in conference proceedings [3, 5, 7, 12, 13], diploma projects [11, 10] and technical reports [4, 9]. During the mentioned works on the MK robot the following topics were studied:

- the idea of driving system,
- methods of measuring the movement parameters,
- open, freeley programmable controller concept.

Two parallel wheels of MK cart are driven by two DC motors mounted on the robot body, which, in turn, is hanged on the axle connecting the wheels. The body forms a kind of pendulum with no additional support points against the ground. Such a construction requires that the control algorithms take into account dynamic properties of the robot.

In order to measure the angle of declination of the body from the vertical axis a two axis accelerometer (ADXL202 from Analog Devices) was used. It has the resolution of 5mg and range of ±2g. Additionaly, the angular velocity of the pendulum is measured with a piezoelectric gyroscope (ENC03J from MuRata). It has the range of $\pm 300\ ^o/$ s.

The robot control system has open architecture and allows easy programming of new control algorithms. All the control functions are performed onboard. The communication with host computer (and, possibly, operator) is used only for setting the trajectory parameters and collecting the measurements for visualization and analysis purposes. It also allows off-line modifications of the controller program stored in FLASH memory.

The software uses the concept of open, freely programmable controller [1], which enables the user to write (in the C programming language) procedures realizing his/her own control algorithms. The controller kernel procedures provide an access both to the measurements and to the control signals, thus freeing the user from knowing the controller hardware details. In order to program a control algorithm, one has to provide a procedure defining and initializing the

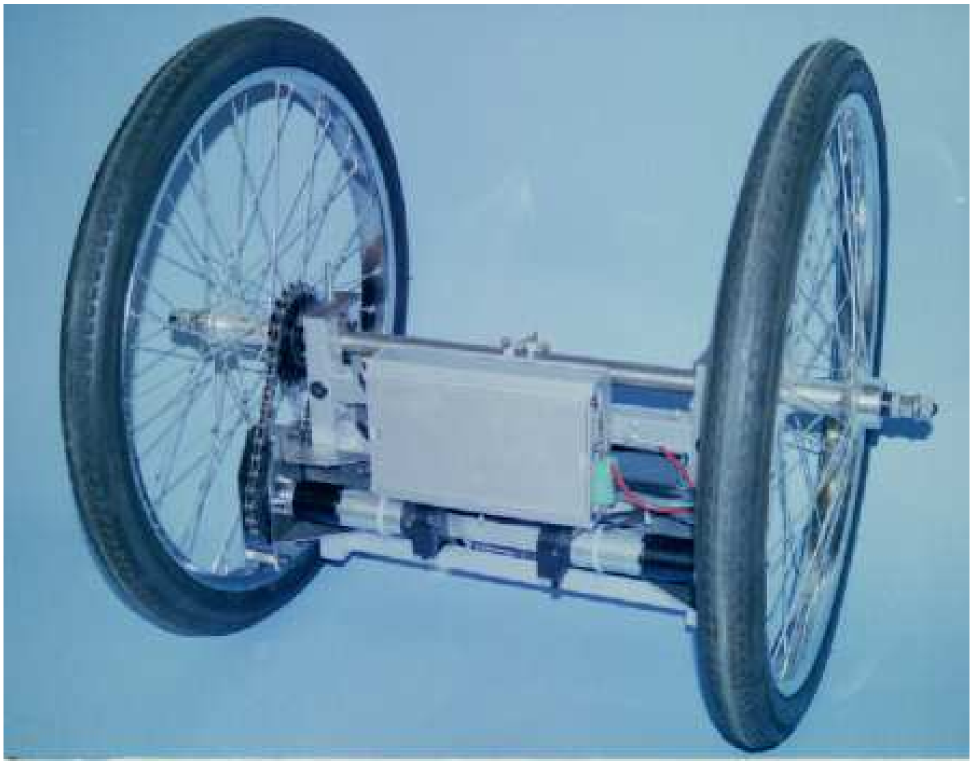

Figure 1. The MK cart

working variables, a procedure calculating the control signals basing on current measurements (one-step control function) and (eventually) define MODBUS registers functions for robot-host communication.

1.2 Modifications following the experimental results

The experiments with MK robot showed the necessity of some improvements of the robot construction:

- using a more powerful, floating point processor in the controller,
- improving the acceleration and angles measurements methods.

MC68332, a 32 bit microcontroller from Motorola, used in MK robot controller, is not equipped with hardware floating point unit, which results in problems in effective implementation of elaborated, model-based control algorithms [7, 11]. A more powerful unit, capable of hardware floating point operations on double prcision numbers would strongly improve the controller efficiency.

Measuring the declination of the pendulum from the vertical axis was based on two perpendicular components of acceleration. The gyroscope was used only for determining the sign of the pendulum rotation. The measurements required extensive filtering to avoid the discretization errors. On the other hand, integrated angular velocity (from the gyroscope) does not require filtering at all. The new concept of the declination measuring system consists in using the integral of the gyroscope signal as the angle measurent and occasionally correcting the angle offset with accelerometrs measurements.

Figure 2. The robot RoBik.

A new version of two wheel robot (called RoBik) was constructed in order to test the suggested enhancements (figure 2). The robot's hardware, software and some experimental results have been described in [19, 20].

2 Mechanical construction of RoBall

The RoBall robot shell is a sphere (actually a globe) which contains power supply, drives (motors and gears) and control system. The RoBall prototype view with one hemisphere of the shell removed is shown in figure 3.

Table 1. Basic mechanical parameters of RoBall

Sphere diameter	0.27 m
Total mass	1.2 kg
Drive module mass	0.45 kg
Accumulators mass	0.3 kg
Pendulum moment of inertia	0.003 kgm^2
Driving torque (per axle)	0.375 Nm
Mechanical power (per axle)	5.22 W
Max. angular velocity	13.9 rad/s

The mechanical drawings and a view of mechanical componets are presented in figure 4. On the sphere poles sleeves are fixed, which connect motor axles with the sphere. Two motors

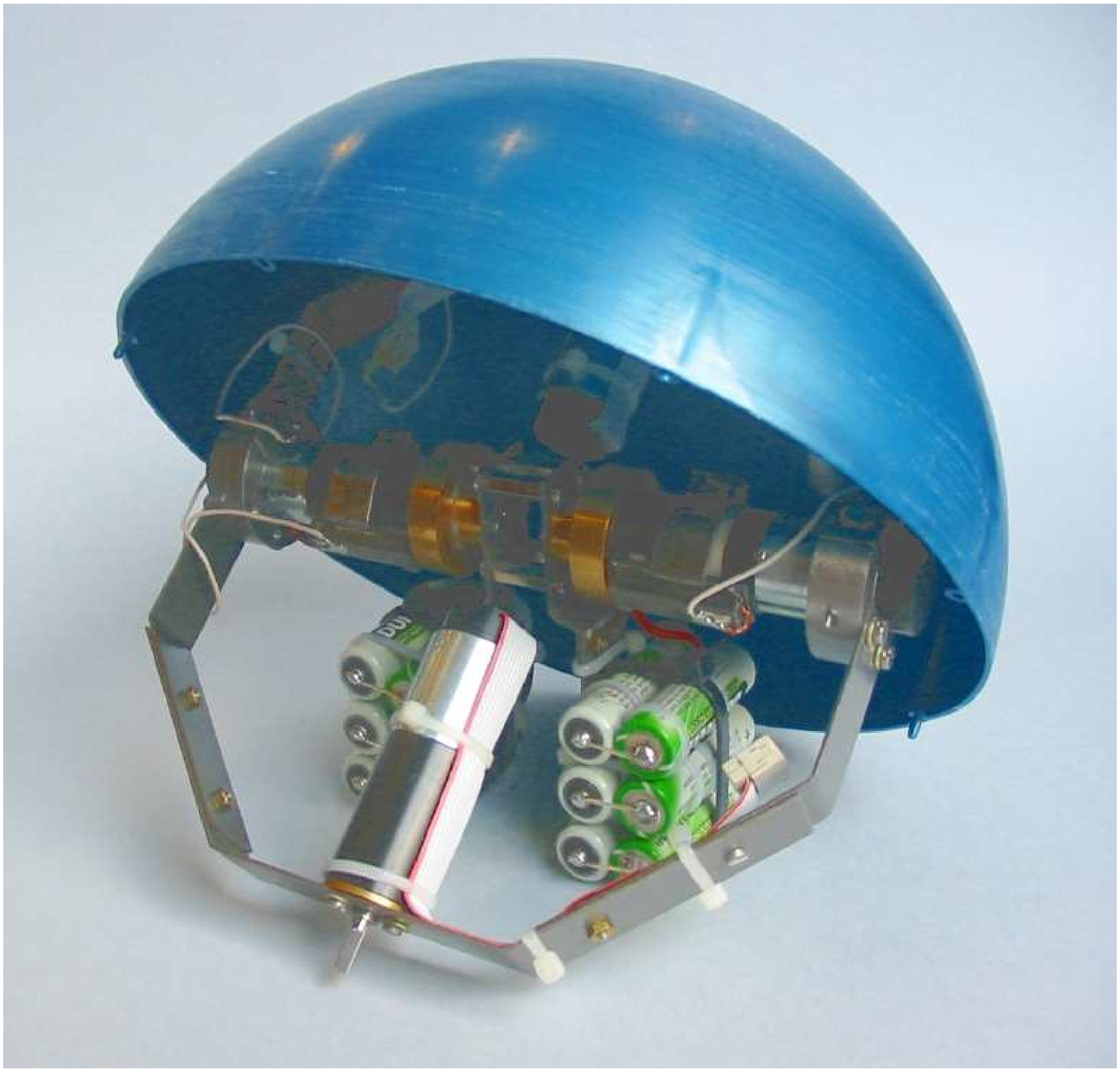

Figure 3. The RoBall overview

working in parallel are used in order to facilitate balancing of the construction. The motors are fixed to a rigid frame (1) (figure 5). Two more motors (3, 11) are fixed with clamping rings (2) to the frame along the axis perpendicular to the axis of the first pair of motors so that their axles meet near the centre of the sphere. A pendulum (7), containing power supply (9) (12 AA-size rechargeable batteries), accelerometers (8), angular velocity sensors, and a local controller providing the measurement subsystem hang on the second pair of motor axles.

All the motors and main electronic modules are fixed to the frame. The current is supplied from the pendulum to the frame via two insulated slip-rings (6) and brushes (4, 5) mounted on the second pair of motors.

The driving torque is provided by moving the mass of the pendulum (mostly accumulators) with respect to the robot shell. It has both static (gravity based) and dynamic (inertial) components. In particular, it is envisioned that one can change the mass distribution of the pendulum in order to compensate the static component.

Following an encouraging experience from MK cart construction, Maxon drive modules

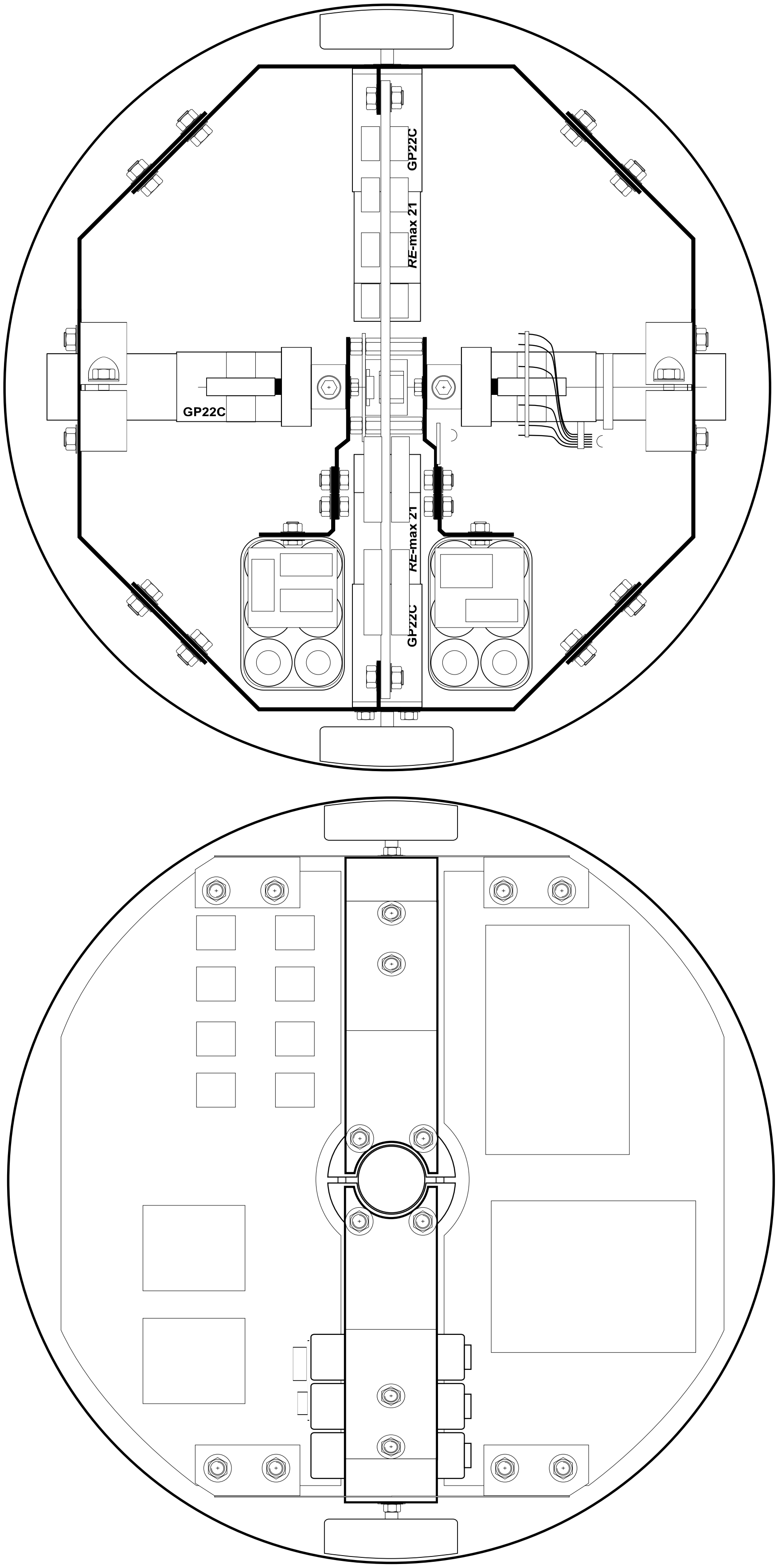

Figure 4. The RoBall construction

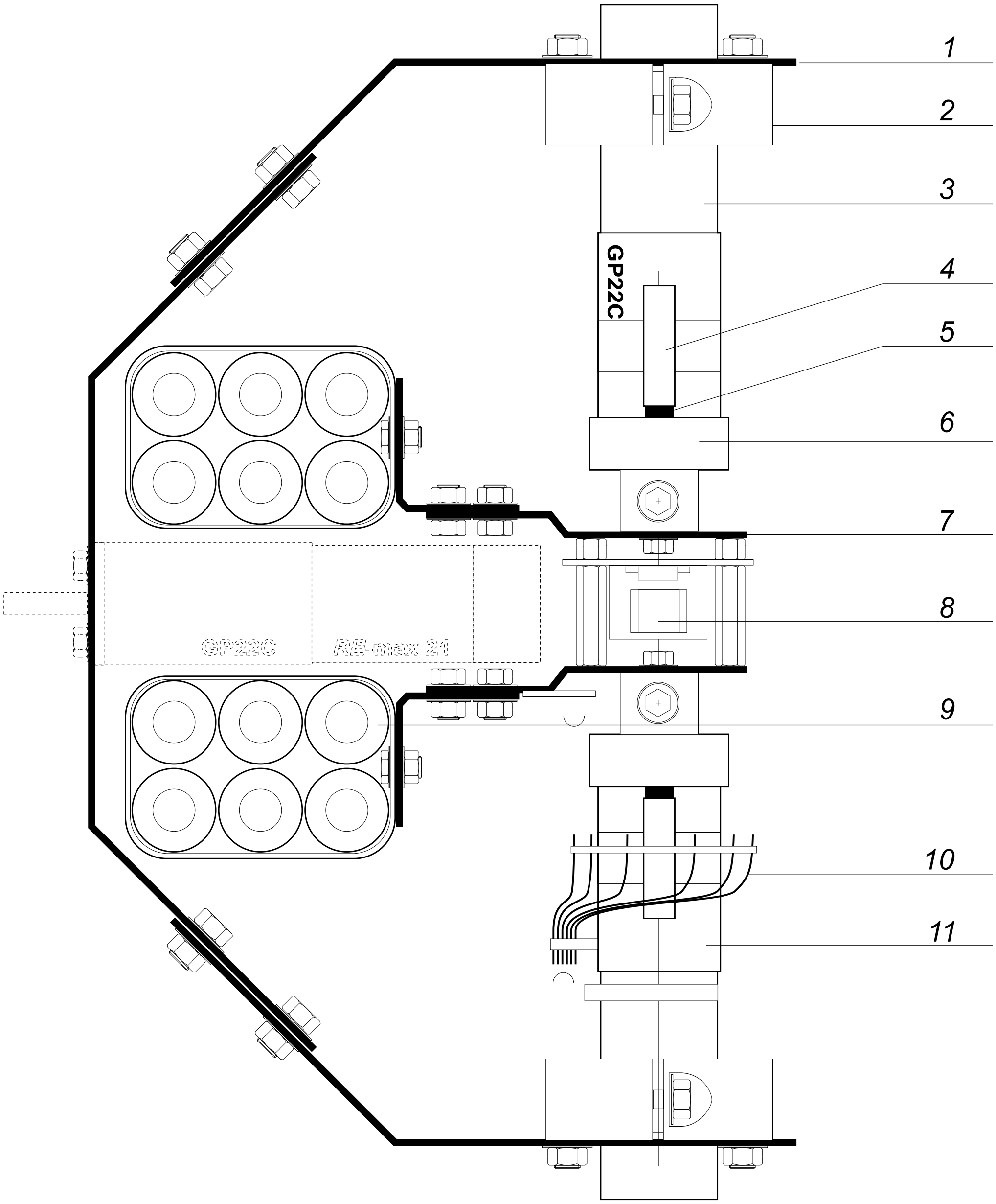

Figure 5. The RoBall mechanics

are used in RoBall. Each of four modules consists of a DC motor RE-max 21 (table 2), 3-stage planetary ceramic gear GP22C (1:53), and incremental encoder MR (512 div/rev).

The mechanical power provided by the used drives allows rotation of the pendulum with the velocity of 2 rev/sec. 12 AA-size NiMH accumulators (2.05Ah each) guarantee more than one hour continuous work of the robot. Two separate DC/DC converters are designed to provide power supply for electronic modules of RoBall. One of them, mounted on the pendulum, provides local controller, accelerometers and gyroscopes with +5V. The other one, mounted on the frame, produces ±5V, +3.3V, and +4V for the main controller module, control logic of the power stages, and wireless communicatiom module.

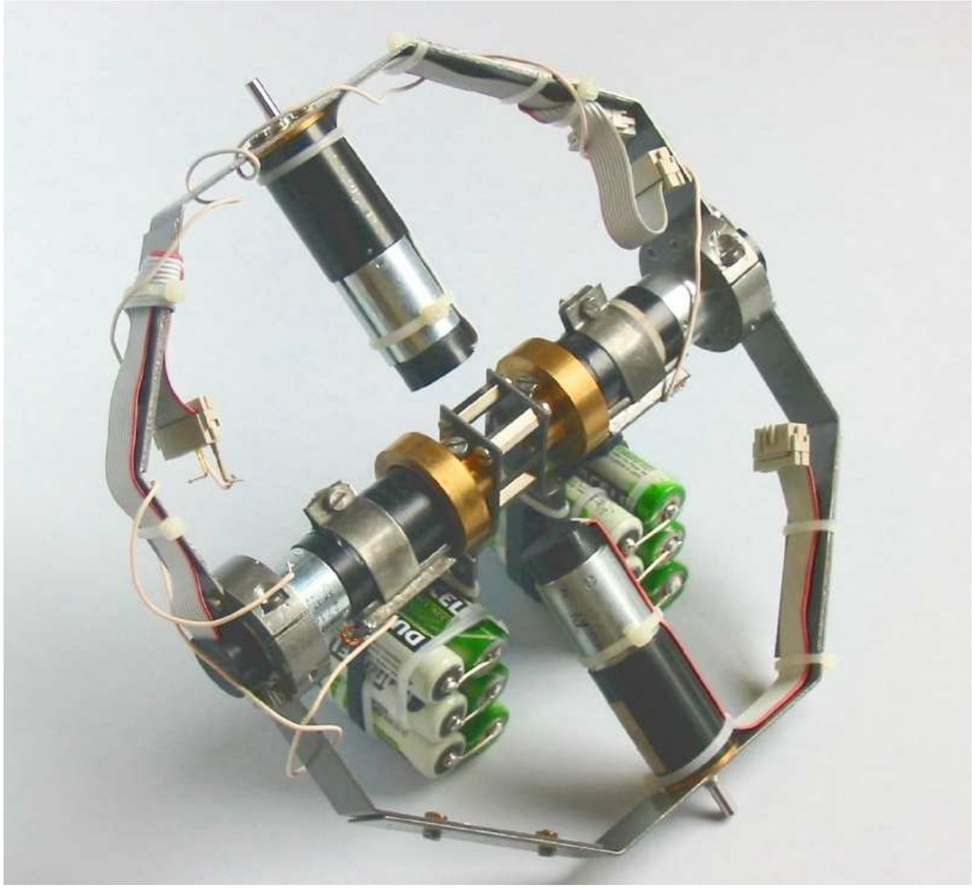

Figure 6. The RoBall frame and pendulum

Table 2. RE-max 21 motor parameters

Assigned power rating	5 W
Stall torque	27.5 mNm
Max. continuous torque	6.8 mNm
Rotor inertia	$0.2 \cdot 10^{-6}$ kgm^2
No load speed	8600 rev/min
Max. permissible speed	10600 rev/min
Mechanical time constatnt	7 ms
Weight	0.042 kg
Nominal voltage	12 V
Terminal resistance	5.77 Ω
Starting current	2.08 A
Max. continuous current	515 mA
No load current	11 mA
Max. efficiency	86 %

3 RoBall control system

The robot is equipped with a control system providing fully autonomous operation. All the sensing, control and effector functions are performed locally. The controller provides means

of communication with operator/host computer in order to set movement parameters, collect experimental data (registered during the movement), and (potentially) change the controller program in FLASH memory.
The control system consists of two separate blocks (see figure 7):

main controller mounted on the frame (MPC555)

local controller mounted on the pendulum (MC9S12C32)

The functional blocks of the main controller are:

- main microcontroller unit,
- power stages with voltage/current control ability,
- incremental encoders readout,
- A/D and D/A converters,
- wireless communication module,
- internal IrDA communication link,
- DC/DC converters (power supply).

Main functional blocks of the local controller are:

- local controller unit,
- three piezoceramic gyroscopes,
- two two-axis accelerometers,
- internal IrDA communication link,
- DC/DC converter (power supply).

Most of the electronics has been placed on two PCBs mounted on the robot frame in the plane perpendicular to the pendulum axis in the centre of the sphere (see figure 4). Accelerometers, gyroscopes and local sensor controller are placed in the centre of the pendulum, on a small PCB.

3.1 Main controller

PhyCORE-MPC555 module from PHYTEC is used as the main controller. It is piggy-backed on the robot PCB with two high density 160-pin connectors. Parameters of the module are summarized in table 3.

MPC555 used in phyCORE-MPC555 is an advanced microcontroller with the following features:

- 32-bit RISC CPU:
  - PowerPC core (52.7 K Dhrystone (v.2.1) @ 40 MHz),

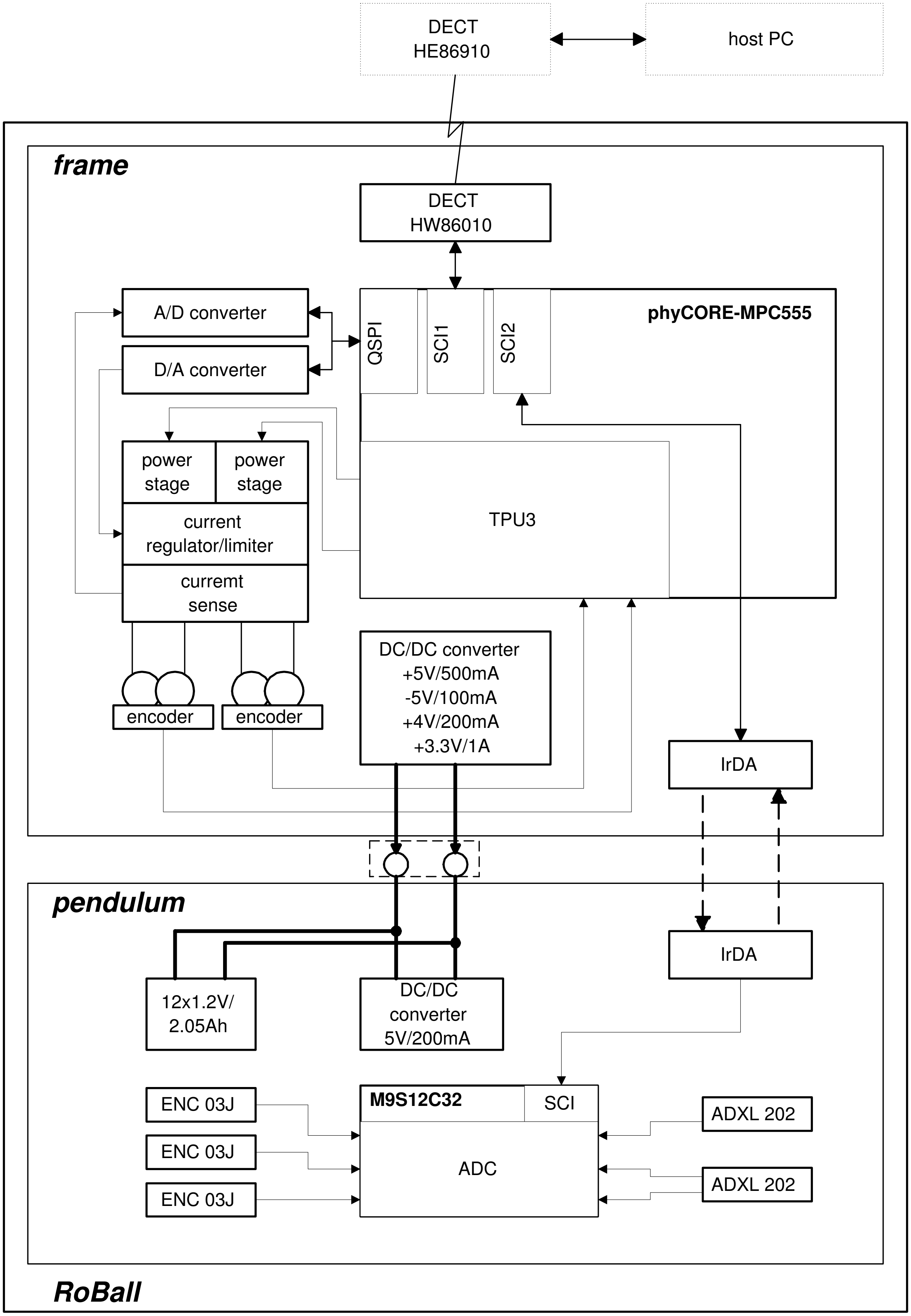

Figure 7. The RoBall controller hardware

  - 64-bit Floating Point Unit,
  - JTAG/BDM test/debug port (OnCE);
- Internal memories:
  - 26kB SRAM + 6Kb TPURAM,
  - 448kB FLASH EEPROM;
- Serial transmission modules (QSMCM):
  - QSPI – 32 slot queue, 8-16 bits,
  - 2 × SCI – external clock capability;
- 2 × CAN 2.0B (TouCAN);
- Timers/counters:
  - 2 × TPU3 – 32 microcoded channels,
  - MIOS1 – 8 × PWMSM;
- 32 10-bit analog inputs – 2 × QADC64.

The interfaces of MPC555 include all the interfaces available in MC68332 microcontroller (used in MK cart) and the microprocessor core is much more effective (particularly in FP arithmetics), what facilitates the implementation of control algorithms.

Table 3. PhyCORE–MPC555 module parameters

Dimensions	ca.75 mm x 57 mm
CPU	32-bit MPC555/40 MHz (272-pin BGA)
Memory	up to 4 MB SRAM, up to 4 MB Flash-ROM
Options	I^2C-EEPROM, I^2C RTC
Interfaces	CAN 2.0B, QSPI, 2× UART (RS-232/TTL)
Converter	2× 16 chan. A/D 10-bit
Debugger	BDM
Power supply	5 V/50mA, 3.3 V/400 mA
MTBF	776765 hrs.

Power stages driving the DC motors are based on H-bridge scheme. Complementary power MOSFETs are used, driven by the logic ensuring safe direction switching (with no cross-over effects). PWM modulation is used to provide current control. The internal current control loop compares the instantaneous current value measured on two resistors with the required current value given by 12-bit DAC. The difference provides the modulation of the H-brigde switches “on” time. The current direction is defined by the polarity of DAC output. Actual current values are measured with 12-bit ADC. Both the DAC and the ADC are connected to MPC555 via QSPI serial bus. An external PWM signal is used to set the average voltage powering the motor. In the case of voltage (velocity) control, current control loop works as current limiter (DAC should be set to max value). In the case of current control, PWM should be set to 100%.

Incremental quadrature encoders mounted on the motors produce signals for motors position and velocity measurement. In TPU3 (Time Processing Unit), an elaborated timer block in MPC555, there are two functions for these measurements: QDEC (Quadrature Decode) and QDVEL (Quadrature Velocity). QDEC is a standard TPU function, which realizes up/down counting of pulses using two input channels (primary - A and secondary - B) for movement direction sensing. QDVEL is a new function in TPU microcode (designed specially for our purposes), which measures rotation speed by counting pulse train period. It uses the secondary channel of corresponding QDEC function to determine the direction.

Table 4. HW86010 DECT module parameters

Dimensions	ca.53 mm x 37 mm x 8 mm
Weight	20 g
Temperature range	-10 to +55 řC
Power supplies	3.0 – 3.6 V digital part, 3.3 – 4.7 V RF part
Current consumption	100 – 150 mA
Frequency	1.88 – 1.9 GHz
Transmitter power	250 mW max.
Antenas	2 internal, 50 Ω connector for external
Range	300 m outdoor, 50 m inhouse
Transmission speed	115 kb/s
Data interfaces	RS-232, 3.3V, up to 115.2 KBd, RTS/CTS
Modem lines	RTS/CTS, DTR/DSR, DCD, RI
Additional interfaces	binary i/o, microphorne, loudspeaker, PCM, I^2C

In MK cart, a bidirectional (half-duplex) wireless link was employed, based on RF modules BiM-2 from Radiometrix, that used a modified MODBUS protocol [8]. Maximum available baudrate was 19.2 kBd. Wireless communication between RoBall and host computer is guaranteed by DECT modules (HW86010 from Hoeft & Wessel). The modules are equipped with a fast, full-duplex serial communication channel allowing for up to 115 kBd transmission rate. The asynchronous interface with modem lines can work in both DTE and DCE configuration. Basic parameters of the module are presented in table 4. The module is mounted on robot frame together with a small board containing level shifters for the interface signals.

At the host side, DECT Evaluation Kit HW86910 is used. It contains two demo boards equipped wit HW86010 DECT module, power supply, RS232 DCE/DTE interfaces and LEDs. After connecting to asynchronous serial port of the host computer the board becomes a fully functional DECT modem.

3.2 Local controller

Local controller contains MC9S12C32 microcontroller. M68MOD912C32 module from Technological Arts is used in the RoBall prototype. The main features of the microcontroller module are summarized in table 5.

Table 5. M68MOD912C32 module parameters

Dimensions	ca.43 mm x 19 mm
CPU	16-bit 9S12C32/24MHz (48-pin LQFP)
Memory	2 kB SRAM, 32 kB Flash-ROM
Interfaces	CAN 2.0, SPI, SCI (RS-232)
Converter	8 chan. A/D 10-bit
Debugger	BDM
Power supply	5 V/25mA

The measurement of the robot accelerations requires that sensors are mounted in the centre of the sphere, on the pendulum. One 2-axis accelerometer is mounted directly in the geometric centre of the robot in the plane defined by the axis of the pendulum and its centre of mass (see figure 5). The second accelerometer lies on a plane perpendicular to the axis so that it measures the third acceleration coordinate. Actually, in order to compensate the parasitic acceleration caused by offset from the centre of robot shell, one should use two accelerometers mounted parallely on both sides of the central point. The accelerometers are of type ADXL202 (range: ± 2g, accuracy: 0.005g). Analog output signals are measured with 10-bit ADC of MC9S12C32 microcontroller. Angular velocities of the pendulum are measured with three gyroscopes of type ENC-3J ($\pm 300\ ^o/$ s) mounted on the pendulum parallely to the accelerometers. The gyroscopes are insensitive to linear accelerations and provide analog signals measured by the mentioned above ADC.

All the measurements are continuously transmitted by local MPU to the main controller by means of an IrDA serial link between SCI module of M9S12C32 and SCI2 module of MPC555.

4 Controller software

In RoBall a concept of open, freely programmable controller [1] has been employed. It guarantees a user the ease of programming his/her own control algorithms for the robot. The structure of controller software is shown in figure 8.

The controller kernel contains procedures and data structure definitions which enable the robot user to implement his/her own algorithm with no need of detailed knowlege of the robot's hardware and control system. The main parts of the controller kernel are:

- **Initialization of the kernel** (hardaware and global data structures);
- **Serial communication service** (robot/host communication in MODBUS standard);
- **Periodic interrupt service** consisting of:
  - **Inputs readout** (signals measured by the sensors),
  - **User algorithm call** (single control step function provided by the user),

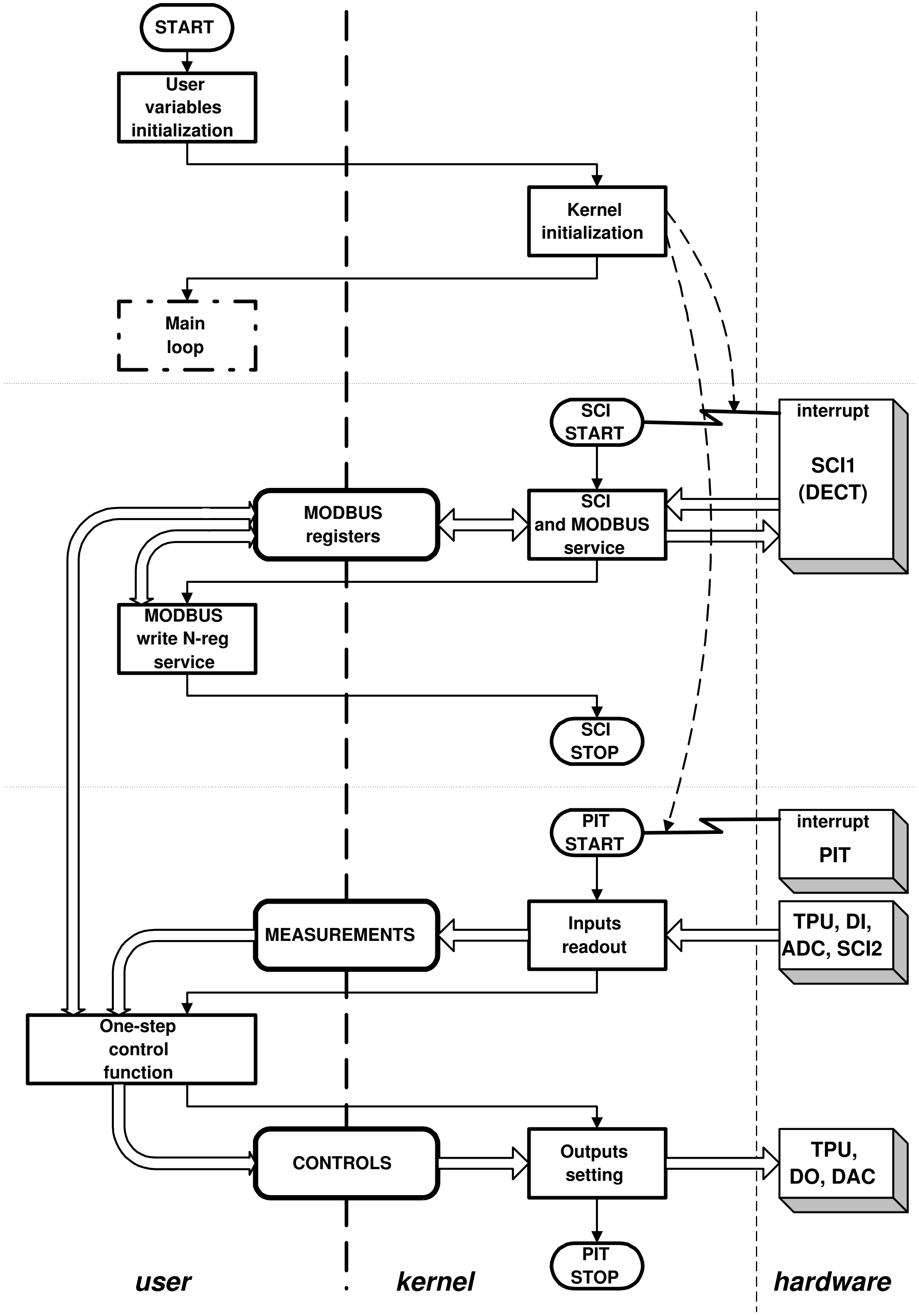

Figure 8. Freely programmable controller software structure

  - **Outputs setting** (calculated control signals).

In order to program user's own control algorithm he/she has to provide (write in the C programming language) the following modules:

- **User variables initialization** (working variables used in user algorithm implemntation),
- **Main loop** (auxiliary procedures - not always required),
- **One-step control function** (a procedure calculating new control signals values on the basis of current values of sensor signals, in accordance with the algorithm being implemented),
- **MODBUS write N-registers service** (providing the means of reaction to asynchronous event of remote parameter modification from the host).

Data structures used for communication between the user code and the kernel are the following:

- **MEASUREMENTS** - current values of the measured and calculated parameters (speeds, angles, currents, etc.),
- **CONTROLS** - current values of control signals (currents, voltages, etc.) calculated by single step control function,
- **MODBUS registers** - a user-defined array of integers for communication with the host (control parameters, motion commands, measured values, etc.).

A preparation of the complete controller program consists in compilation of user's modules, linking the resulting objects with kernel modules, and loading the binary code to on-board FLASH memory (in the RoBall prototype, the external FLASH of the phyCORE module is used). After reset the new software starts automatically.

Note that the local controller (MC9S12C32) is not expected to be user-programmable. It transmits the measurements from the gyroscopes and the accelerometers mounted on the pendulum via IrDA local link to the main controller in a precisely defined form. The IrDA communication procedure at the main controller side is contained in the kernel code. It updates the appropriate fields of **MEASUREMENTS** structure on the basis of received data.

As a cross-development environment for both the MPC555 and MC9S12 microcontrollers Code Warrior from Metrowerks is used. CodeWarrior is a commercial software package available for many microcontrollers. It can be hosted on PC computer with Windows, but Solaris and Linux versions are also provided. The package contains IDE with editor and project manager, assembler, linker, source-level debugger, fast simulator, code profiler, and FLASH memory programmer. The debugger makes use of BDM (Background Debug Mode) interface embedded in Motorola microcontrollers. BDM enables the user to debug both the hardware and the software in-system, with no additional hardware emulators. The cheap interfaces between PC host and the target system use a parallel printer port (Wiggler for MPC555) or a serial port (SDIL for M9S12).

An alternative, prospective solution is using GNU toolchains for PowerPC and CPU12 processors, freely available in accordance with GPL (General Public License) rules. The toolchain contains GCC C/C++ compiler, assembler, linker amd GDB debugger. There are GDB patches available, which allow using BDM for in-system debugging.

5 Concluding remarks

A nonholonomic, spherical mobile robot turned out to be a very interesting object for testing new ideas dealing with modelling the dynamics and kinematics of nonholonomic robots. As an object of control it is quite sophisticated. Its behaviour remains beyond the intuition and is hard to imagine without experimenting. Moreover, simulations [14] show some strange effects, hard to be explained. The described spherical robot RoBall was developed as a physical object, which can be used in experiments including, but not limited to, model verification, parametric identification, control algorithms design and testing. Its immanent feature is the necessity to consider robot dynamics even while solving kinematic tasks.

Using simulation techniques one tends to simplify the real world to obtain a readable, “clean” model. This results in possibility to overlook phenomena which are essential for the real object anlysis. Using a real object instead of a model allows to avoid such simplifications, thus making the research results more adequate. The spherical robot RoBall is equipped with an open, freely programmable control system, which can be easily reprogrammed by a user, who does not have to understand in-deep the controller hardware. MPC555 microcontroller with PowerPC core, hardware floating point unit, two TPU3 microcoded timers, serial interfaces, and AD converters seems to be a good choice for control algorithm implementation. Using a cross-development package one can prepare programs in C language and load them in FLASH memory of the controller, thus implementing user’s own control algorithms. All the hardware service routines are prepared in form of kernel (object code), which is linked with the user routines.

One of the envisioned directions of the project extention consists in modifying the RoBall drive by compensation of the static (gravitation based) torque component. After such a modification, RoBall’s behaviour should change significantly. From the technical point of view, the mass distribution of the pendulum is easily changed by symmetrical placement of the batteries.

A simplified version of RoBall drive may be used in future modifications of an interactive, spherical robot for the therapy of authistic children, being developed in the Unit of Foundations of Cybernetics and Robotics [16, 18, 17].

References

[1] S. Gora, M. Wnuk *VFMC - Freely programmable robot motion controller*. Report SPR 13/97, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 1997 (in Polish).

[2] R. Muszyński, K. Tchoń. Singularities and mobility of nonholonomic systems: the ball rolling on a plane, In *6th IFAC Symposium on Robot Control, SYROCO'00*, Vienna 2000, Preprints vol.1, pp. 259-264.

[3] M. Kabała, K. Tchoń, M. Wnuk. A mobile robot driven in internal coordinates. In *VII KKR, Lądek Zdrój, 2001*, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, Conferences 46, t.1, pp. 149-158 (in Polish).

[4] M. Kabała, M. Wnuk. *Two wheel mobile robot driven in internal coordinates*. Report SPR 21/2001, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2001 (in Polish).

[5] M. Kabała, K. Tchoń, M. Wnuk. Two wheel nonholonomic mobile robot. In *Automation 2002, Conference proceedings*, Warszawa, 2002, pp. 269-280 (in Polish).

[6] M. Kabała, M. Wnuk. *A spherical mobile robot RoBall (preliminary project)*. Report SPR 37/2002, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2002 (in Polish).

[7] K. Tchoń, M. Kabała, M. Wnuk. An algorithm for trajectory tracking for MK mobile robot. *XIV KKA, Conference proceedings*, Zielona Góra, 2002, (in Polish).

[8] M. Wnuk. *Wireless communication with MK mobile robot*. Report SPR 19/2002, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2002 (in Polish).

[9] M. Kabała, M. Wnuk. *Construction and programming of two weel mobile robot*. Report SPR 20/2002, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2002 (in Polish).

[10] M. Kabała. *A mobile robot control based on its dynamics*. MSc.thesis, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2002 (in Polish).

[11] C. Wronka. *Kinematics and dynamics model of a nonholonomic robot*. MSc.thesis, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2002 (in Polish).

[12] M. Kabała, R. Muszyński, M. Wnuk. Singularity robust, dynamic linearization control algorithm for MK mobile robot, In *Robot Control 2003 (SYROCO'03)*, Elsevier Ltd, 2003, vol.2, pp. 467-472.

[13] R. Muszyński, C. Wronka. MK mobile robot with simplified controller. In *Robot Control 2003 (SYROCO'03)*, Elsevier Ltd, 2003, vol.2, pp. 473-478.

[14] M. Kabała M, W. Matysek, R. Muszyński. Dynamics model of spherical robot RoBall with nonholonomic constraints. In *Progress in robotics: robot control* (in print).

[15] M. Wnuk. Measuring the dynamic parameters of two wheel mobile robot. In *Progress in robotics: industrial and medical robotic systems* (in print).

[16] K. Arent. User interface of s spherical robot supporting therapy of authistic children. In *Progress in robotics: robot control* (in print).

[17] M. Kabała. Sensors and effectors in a spherical robot supporting therapy of authistic children. In *Progress in robotics: robot control* (in print).

[18] M. Wnuk. A user-programmable controller for a sphericalrobot supporting therapy of authistic children. In *Progress in robotics: robot control* (in print).

[19] R. Szlawski, M. Wnuk. *Construction of two weel mobile robot RoBik*. Report SPR 12/2004, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2004 (in Polish).

[20] R. Szlawski. *Implementing and testing of two wheel robot control algorithms*. MSc.thesis, Inst. of Eng. Cyb., Wroclaw Univesity of Technology, 2004 (in Polish).

mgr inż Marek Kabała
dr inż. Marek Wnuk
Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej
ul. Janiszewskiego 11//17
50-372 Wrocław

Niniejszy raport otrzymują:

1. OINT .... - 1 egz.

2. Zleceniodawca .... - 2 egz.

3. Autorzy .... - 2 egz.

Razem : 5 egz.

Raport wpłynął do redakcji I-6
w październiku 2004 roku.