Chapter 2
ROBOTIS' Robot Systems

2.1 General Systems Description

ROBOTIS Co., Ltd. (Seoul, South Korea) was founded by Bill Byoung-Soo Kim in 1999 along with two other engineers. The current CEO (Byoung-Soo Kim) and Vice-President (In-Yong Ha) were from the original team. The ROBOTIS name derived from the question "What is a robot?" and their vision statement can be read at (http://en.robotis.com/index/company_01.php). In 2009, ROBOTIS opened their USA office in Irvine, California and in 2016, the USA office moved to Lake Forest, CA. Currently ROBOTIS has more than 200 partners in 40 countries worldwide.

"Dynamixel" was the brand name uniquely connected to ROBOTIS. "Dynamixel" encapsulated several modularization and standardization concepts applied to both sensors and actuators equipped with embedded computing and communications capabilities (see Fig. 2.1). Actually, the Main Hardware Controller was also considered as a Dynamixel with a reserved ID = 200, and a co-controller such as a Smart Phone would have a reserved Dynamixel ID = 100 (see Chap. 12 for more details).

In 1999, ROBOTIS launched their first product called Didi and Titi (see Fig. 2.2). This web link shows a TV commercial for Didi and Titi (http://en.channel.pandora.tv/channel/video.ptv?c1=05&ch_userid=do7minate&prgid=49759295&ref=na).

Since then, ROBOTIS has released 27 more products:

1. Toma (2002 – released in Korea only).
2. Dynamixel – AX-12 (2003).
3. Cycloid (2004 – released in Korea only).
4. BIOLOID – Beginner and Comprehensive (2005).
5. URIA (2006 – released in Korea only).
6. Dynamixel – RX-64 (2006).
7. OLLO (2008).
8. Dynamixel – EX-106 (2008).
9. BIOLOID PREMIUM (2009).
10. Dynamixel – MX series (2011).

C.N. Thai, *Exploring Robotics with ROBOTIS Systems*,
DOI 10.1007/978-3-319-59831-4_2

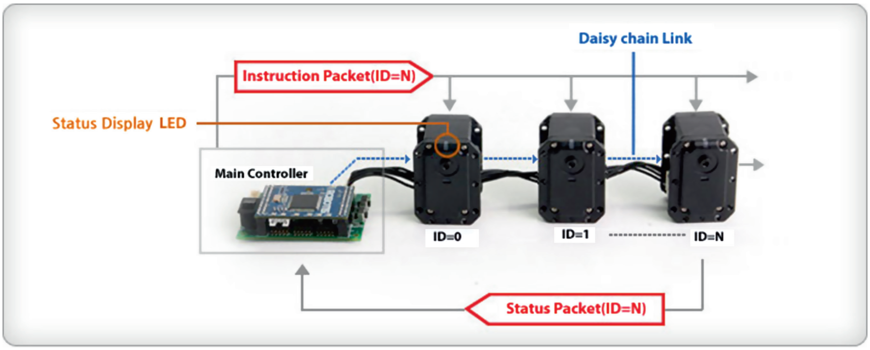


Fig. 2.1 "Dynamixel" concept (Courtesy of ROBOTIS)

Fig. 2.2 Didi and Titi (Courtesy of ROBOTIS)

11. DARWIN-OP (2011), renamed ROBOTIS-OP in 11/2014.
12. BIOLOID STEM (2012).
13. IDEAS (2013).
14. THOR-MANG (2013).
15. Dynamixel-Pro H-series (2013).
16. Dynamixel – XL-320 (2014).
17. ROBOTIS MINI and OpenCM-9.04 (2014).
18. Dynamixel-Pro L-series (2014).
19. DREAM and PLAY (2014).
20. SMART I (2015) and SMART II (2016).
21. IoT (2016).
22. OLLOBOT (2016).
23. Dynamixel X-series (2016).
24. TurtleBot3 and OpenCR (2016).
25. PLAY 700 system (2016).
26. SMART III (2017).
27. MiCom Educational Kit, based on the OpenCM-9.04 (2017).

This list shows ROBOTIS' commitment to continuing development and improvement and to serve a very broad clientele in terms of age as well as technical level.

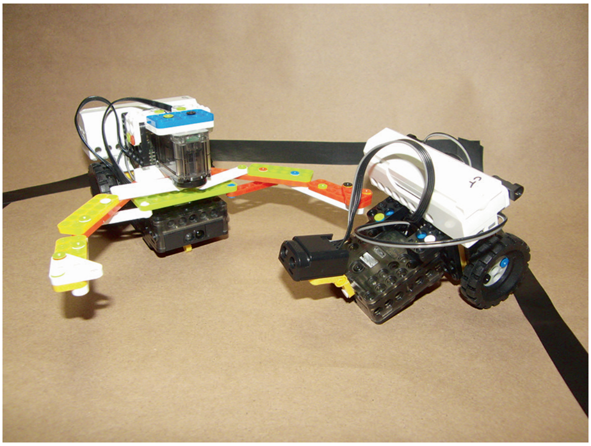


Fig. 2.3 DREAM robots

These products had been adopted by hobbyists of all ages as well as teachers and researchers worldwide (please visit http://www.ROBOTIS.com for more details).

Currently, the systems designed for young children are: PLAY, DREAM, SMART and IoT. They are colorful and use a quick-connect system adapted from the standard rivet concept to ease hands-on creative activities for young children (see Fig. 2.3). They can be constructed to be simple motorized robots (PLAY 200–600 series) as well as programmable robots (PLAY 700, DREAM, SMART and IoT). Furthermore PLAY 700, DREAM, SMART and IoT system were designed to be used with mobile devices such as cell phones and tablets (https://play.google.com/store/apps/developer?id=ROBOTIS&hl=en). Their embedded controllers are based on the STM32F103C8 and STM32L151C8 from STMicroelectronics.

The PLAY 700 system was introduced in late 2016 and it used the CM-50 controller which could be used with either one of two software packages: the MIT SCRATCH V.2 (via the ROBOTIS' R + SCRATCH tool acting as an HTTP extension), or with the ROBOTIS' R + TASK tool for the creation of programs that could execute on the CM-50 directly and interact with a PC or a mobile phone/tablet in either Android® or iOS® operating system (see Fig. 2.4 and playlist at https://www.youtube.com/playlist?list=PLtix7rPAJwqzqvrI6nr4qI0FILYrLqbgb).

The DREAM system used a controller named CM-150, while the SMART systems used the CM-200 (see Fig. 2.5). Although visually similar to each other and both capable to run TASK programs, the CM-200 was slightly larger and had more capabilities than the CM-150 such as eight GPIO ports (CM-200) instead of two (CM-150) and only the CM-200 could additionally support the MOTION programming features provided by the R + Motion V.2 tool (see Chap. 4 for more details on the R + Motion tool). The DREAM system can also use the R + m.PLAY700 Mobile App to access the SMART facilities on the smart phone (see video at https://www.youtube.com/watch?v=-O-vZhAH-bQ and Chap. 11 for more details). The SMART III system has its own Mobile App package and the SMART III system would be

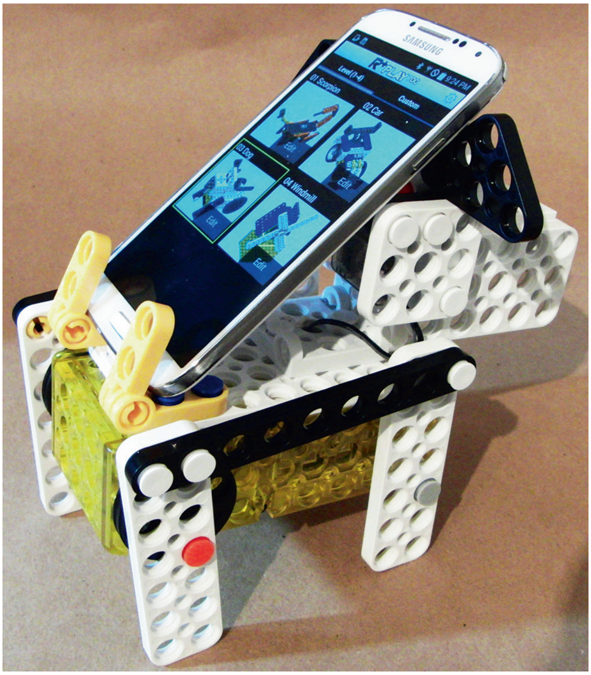


Fig. 2.4 DOG robot from the PLAY 700 series

Fig. 2.5 CM-150 (*left*) and CM-200 (*right*)

suitable for a First-Year Odyssey type of course for freshman engineering students, alas currently this system is only available for Korean speaking users.

The author had used the PLAY 700, OLLO, DREAM and BIOLOID STEM systems to teach introductory robotics to middle and junior high school students via week-long and week-end short courses. The instructional approaches and materials used are described at the following web links:

- Summer Academy at UGA (https://www.cntrobotics.com/sauga)
- Duke TIP Academic Adventures (https://www.cntrobotics.com/duke-tip-aa)
- Duke TIP Scholars Weekend (https://www.cntrobotics.com/duke-tip-sw)

The IoT system used the OpenCM7.0 as controller (i.e. STM32L151C8). Similarly to the CM-50, it could be interfaced either via the SCRATCH V.2 tool from MIT for “IoT Level 1” (see Sect. 4.6 for more details) or via the usual ROBOTIS R + TASK V.2 tool for “IoT Level 2” and its own mobile app in both Android and iOS flavors (see Chap. 4 also for more details on the R + TASK tool). As of late 2016, the IoT systems are only available for Korean speaking users (see Fig. 2.6).

The BIOLOID systems (BEGINNER, COMPREHENSIVE, STEM and PREMIUM) are designed to be various entry points (depending on one’s budget) into the robotics field for those interested in taking a more comprehensive journey into this knowledge area. They use standard screws and nuts for a sturdier fastening method, with some parts using the DREAM/SMART’s rivet system also. The older kits (BEGINNER and COMPREHENSIVE) were designed off the Atmel AVR chip while more recent systems (STEM and PREMIUM) rely on the ARM architecture using such embedded controllers as STM32F103C8 and STM32F103RE from STMicroelectronics (see Fig. 2.7).

In 2012, ROBOTIS made a strategic shift into the open hardware-software movement with their OpenCM systems whereas users can collaborate on hardware and software development worldwide (see Fig. 2.8).

ROBOTIS is also well known for its humanoid robots in the competitive and research arenas such as the ROBOTIS GP (http://en.robotis.com/index/product.php?cate_code=121510), ROBOTIS OP (http://en.robotis.com/index/product.php?cate_code=111010) and ROBOTIS OP2 (http://en.robotis.com/index/product.php?cate_code=111310), and THOR-MANG3 (http://en.robotis.com/index/product.php?cate_code=111410). In Spring 2014, ROBOTIS released a more affordable humanoid robot called ROBOTIS MINI (http://en.robotis.com/index/product.php?cate_code=121310) based on the OpenCM9.04-C controller and the XL-320 servo motor. Outwardly, the ROBOTIS MINI (see Fig. 2.9) is roughly a half-scale version of the ROBOTIS OP, but with reduced capabilities and a different kinematic linkage solution for its legs. It is designed to be operated and programmable from mobile devices (Android® and iOS®) as well as personal computers (MS Windows®). The MINI system can also use the R + m.PLAY700 mobile app to

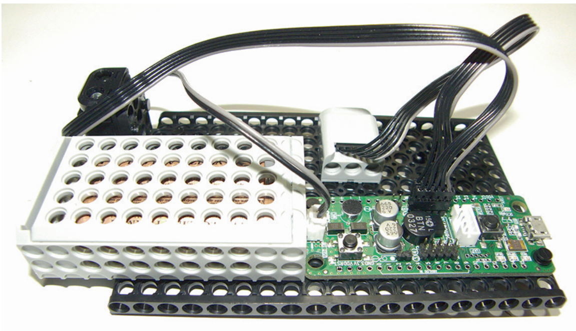


Fig. 2.6 ROBOTIS IoT system (level 1)

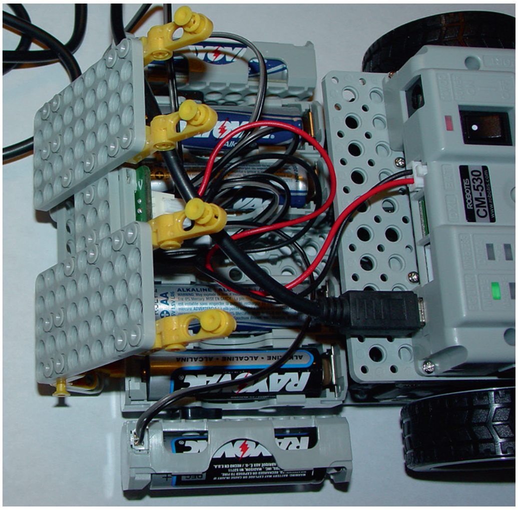


Fig. 2.7 A bioloid STEM robot

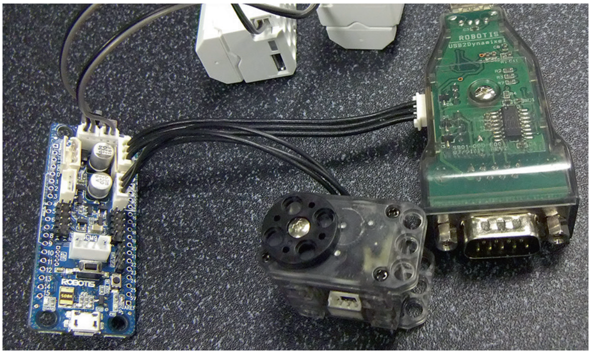


Fig. 2.8 OpenCM-9.04-B controller (*left*), XL-320 servo motor (*center*), and USB2Dynamixel communication converter (*right*)

access the SMART facilities on the smart phone (see video at https://www.youtube.com/watch?v=-O-vZhAH-bQ and Chap. 12 for more details).

The OLLOBOT was introduced to developers at the SHAPE AT&T Tech Expo in July 2016 (see Fig. 2.10). It can be controlled via a cell phone (Android or iOS) with an Android App (https://github.com/ROBOTIS-GIT/OLLOBOT) or with an IFTTT IF® App (https://play.google.com/store/apps/details?id=com.ifttt.ifttt).

The TurtleBot 3 was introduced at the ROSCon 2016 conference for developers and university graduate students who wanted to use ROS™ (http://www.ros.org/) and currently (Spring 2017) it came in two models: "Burger" and "Waffle" (see Fig. 2.11). It used the OpenCR controller and was designed to be modular with an open hardware and software paradigm (http://turtlebot3.readthedocs.io/en/latest/).

Fig. 2.9 ROBOTIS MINI action figure

Fig. 2.10 The OLLOBOT system with a Samsung S4® phone

As this book is geared towards university undergraduate students or self-learners, the following systems will be considered in further details: BIOLOID BEGINNER, COMPREHENSIVE, STEM and PREMIUM, OpenCM, PLAY700, ROBOTIS MINI and OLLOBOT.

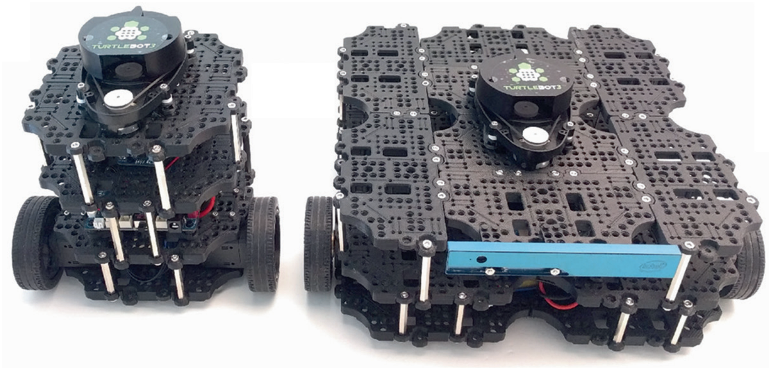


Fig. 2.11 Current TurtleBot 3 models: "Burger" (*left*) and "Waffle" (*right*) (Courtesy of ROBOTIS)

Fig. 2.12 CM-5 controller (discontinued)

2.2 Robotics Kits Considered in Book

The BIOLOID BEGINNER and COMPREHENSIVE systems use the CM-5 (see Fig. 2.12) as its main controller which is an Atmel ATmega™ @ 16 MHz and with 128 KB of flash memory. It connects to ROBOTIS' sensors and actuators via the Dynamixel TTL bus (three-pin) and has ZigBee wireless communications capabilities via the ZIG-100 modules.

Actually there was another CM-5 based system called BIOLOID EXPERT that was available between 2005 and 2009. It used the same basic hardware as the COMPREHENSIVE, but had a wireless color video camera (Fig. 2.13) and a Visual C++ V.6 library with functions to control sensors, actuators, video camera and also to perform machine vision processing routines and ZigBee wireless communications.

When the BIOLOID PREMIUM system first came out in 2009, it was shipped with the CM-510 controller (see Fig. 2.14-*left*) which was a 16-MHz ATmega with 256 KB flash memory, but since 2013 it is shipped with the CM-530 (Fig. 2.14-*right*)

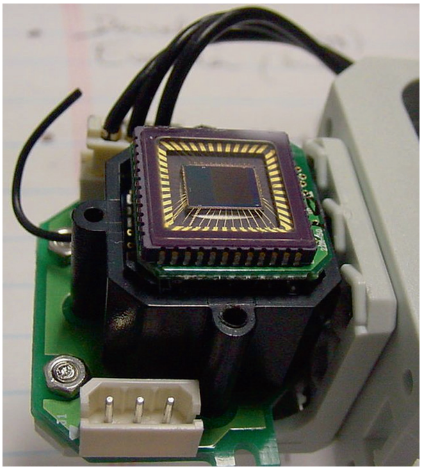


Fig. 2.13 Wireless color video camera from BIOLOID EXPERT kit (discontinued)

Fig. 2.14 CM-510 (*left*-discontinued) and CM-530 (*right*) controllers

which is based on the 72-MHz STM32F103RE with 512 KB flash memory. Otherwise, visually and functionally, the CM-510 and CM-530 are identical to each other. Both are also capable of handling embedded C applications via the WinAVR (CM-510) or WinARM (CM-530) tool chains (see further discussions in Chaps. 4 and 9).

It should be mentioned that there are also two bare bone ATmega-based controllers, CM-2+ (discontinued) and CM-700 (see Fig. 2.15), which were designed for custom needs when the user has to mix two types of Dynamixel modules together in the same controller (i.e. three-pin TTL and four-pin RS-485, see Chap. 3 for more details). Embedded C is also available for the CM-700.

The latest BIOLOID system (2012) is the STEM kit which combines hardware construction approaches from the previous BIOLOID kits (i.e. screws and nuts) and from the DREAM kits (i.e. plate and rivets). The STEM also has new hardware to create more secure pin joints (Fig. 2.16) and an IR Sensor Array (to be described later in Chap. 3). The STEM kit uses the CM-530 controller. It comes as two separate kits, Standard and Expansion, and the Standard kit is required for the proper use of the Expansion kit.

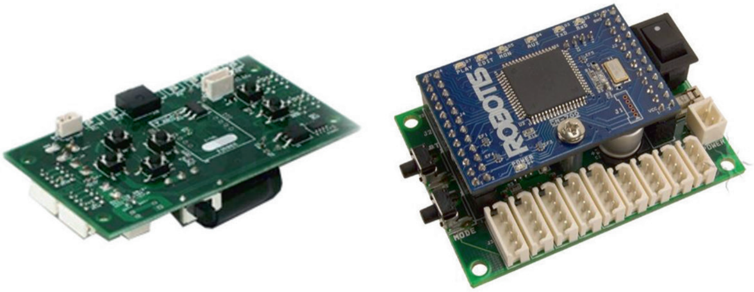


Fig. 2.15 CM-2+ (*left* – discontinued) and CM-700 (*right*) controllers (Courtesy of ROBOTIS)

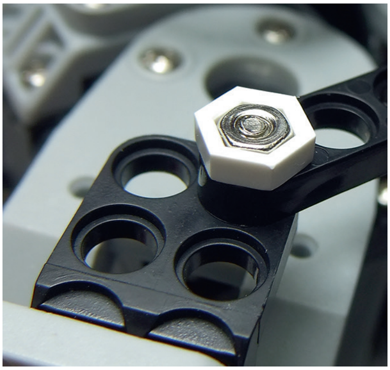


Fig. 2.16 BIOLOID STEM pin-joint hardware

All ROBOTIS BIOLOID systems use the RoboPlus software suite consisting of four tools:

1. MANAGER – for general hardware troubleshooting and obtaining firmware update for controllers and dynamixels.
2. TASK – general programming environment for the user.
3. MOTION – motion programming environment for multi-links robots.
4. DYNAMIXEL WIZARD – for troubleshooting and updating firmware on Dynamixel actuators and sensors.

The MANAGER, TASK and MOTION tools exist in two versions depending on which controllers are being used (see Chaps. 3 and 4 for more details).

Starting in 2014, the OpenCM platform is the vehicle for ROBOTIS to accomplish its open-hardware and software goals in the near future for operating systems such as MS Windows®, Mac OSX® and Linux®. The CM-900 was the beta platform first available in 2012 but is no longer sold by ROBOTIS. It had two editions, ES and V.1.0, which are based on the STM32F103C8 microcontroller (Fig. 2.17). They carried 64 KB of flash memory and support many hardware interface standards such as USB (1), CAN (1), USART (3), I2C (2) and SPI (2). The ES version supported both AX/MX-TTL and RS-485 Dynamixel ports, the V.1 edition additionally supported the new XL-TTL Dynamixel port. Both supported software

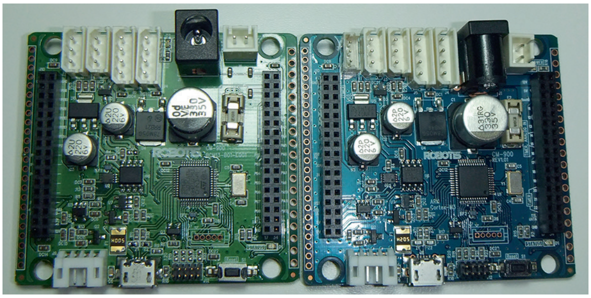


Fig. 2.17 CM-900 ES (*left*) and V. 1 (*right*)

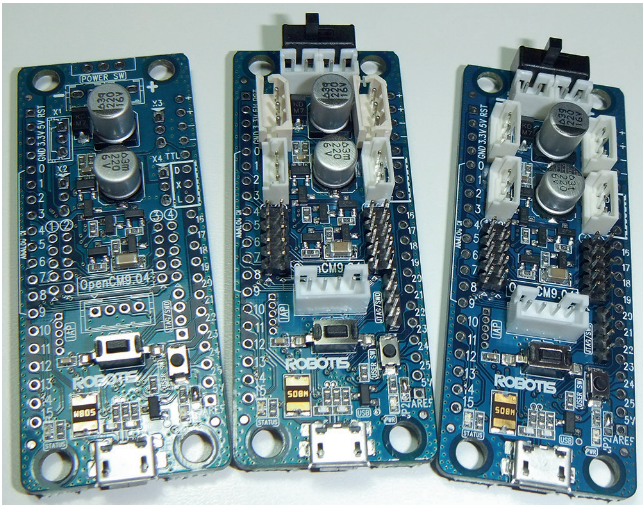


Fig. 2.18 OpenCM-9.04-A/B/C controllers (*left* to *right*)

development using an Arduino-based IDE (called "OpenCM IDE") and wireless communications programming via ZigBee and BlueTooth. Real-time debugging (SWD and JTAG) was also available using additional tools such as ST-LINK/V2 and Keil μVision.

Currently, only the OpenCM-9.04 series is commercially available and it comes in three versions A, B and C (Fig. 2.18). The 9.04 series has the same hardware features previously listed for the 9.00 series, however they have a smaller physical format and 128 KB of flash memory.

The B version is "ready-to-go" if the user plans to use a mixture of AX/MX-TTL and XL-TTL Dynamixel modules (but it is no longer available commercially). The A version is essentially a user-customizable B version whereas the user can install only the needed headers. Both A and B versions are completely open hardware and software controllers whereas users can adapt their own firmware and boot loader as they wish. They use the OpenCM IDE as the programming interface.

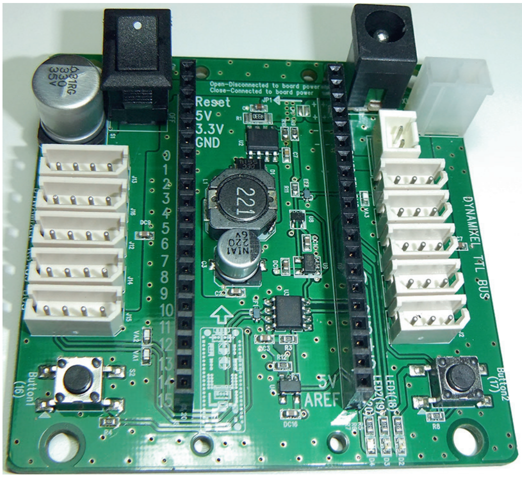


Fig. 2.19 Expansion board OpenCM-485-EXP for the OpenCM-9.04 series (beta version – *green*, release version – *blue*)

Currently the C version comes with the ROBOTIS MINI kit and is also available by itself (http://www.robotis.us/opencm9-04-c-with-onboard-xl-type-connectors/). It has only the XL-TTL connectors installed. Most importantly, it has a proprietary firmware so that it can operate with the R + Task and R + Motion V.2 software suite. Alternately, the user can use the OpenCM IDE with version C but this mode would effectively erase the proprietary firmware, thus if the user wants to use the R + Task and R + Motion packages again, a firmware recovery process must be performed (see Chap. 4).

If the user requires more TTL and RS-485 Dynamixel ports, the expansion board OpenCM-485-EXP can be used (Fig. 2.19).

The CM-50 (PLAY700 system) and OpenCM-9.04/C + 485-EXP combo would also be used extensively to illustrate the SMART structure that ROBOTIS created to let its Firmware 2.0 controllers to collaborate with mobile devices running on Android and iOS systems (see Fig. 2.4 and Chaps. 9 and 11).

2.3 MiCom Training Kit

In 2017, ROBOTIS plans to release the "MiCom Training Kit". It is based on the OpenCM9.04-C and oriented towards the DIY type of user. Figure 2.20 shows its main components:

- One OpenCM-9.04-C with 2-mm header mounted.
- Three NIR sensors (5-pin type).
- Two XL-320 actuators.
- One solderless breadboard and some 100 mm jumper wires.
- Some electrical and electronics components such as resistors (10 KΩ and 100 KΩ), 1 red LED, matching NIR LED emitter/receiver, 1 toggle switch, 1 variable resistor, a microphone and a 7-segment LED display.

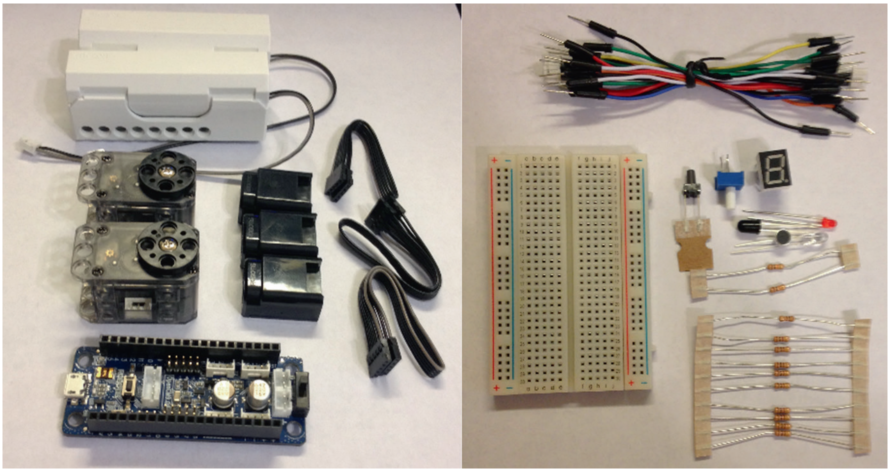


Fig. 2.20 Micom training kit (due in 2017)

- Additional OLLO frame and wheel parts to make a wheeled robot using NIR sensors so that it can follow a black track (for example).

With this kit, the DIY user can start programming using the TASK tool for a quick learning approach, then later can switch to the Arduino based ROBOTIS IDE for a low-level programming style using the C language and the breadboard for experimental circuits such as:

- Direct pin access and digital output (PWM and duty cycle).
- Analog input and A/D conversion.
- Serial communications.
- Dynamixel control and instruction packet management.
- Memory access, etc.…

A review of this system is described in "Video 2.1" and the mentioned Arduino project "MiCom_1.ino" is included in the Springer Extras materials.

2.4 OLLOBOT

Relatively speaking, the OLLOBOT can be considered as the "odd ball" of the ROBOTIS family, but it still uses the familiar ARM controller STM32F103RET6 (72 MHz, 512 KB Flash RAM) – see Fig. 2.21. Please note the four sets of five-pin connector at the top of the circuit board which could be used for GPIO sensor interfaces in the future perhaps?

Figure 2.22 represents the "current" (Fall 2016) Control Table for the OLLOBOT controller listing its various built-in functions that are programmable. The reader can see that only two servo motors are accessible, thus the sensing and high-level logic would have to be done at the cell phone level. So working with the OLLOBOT

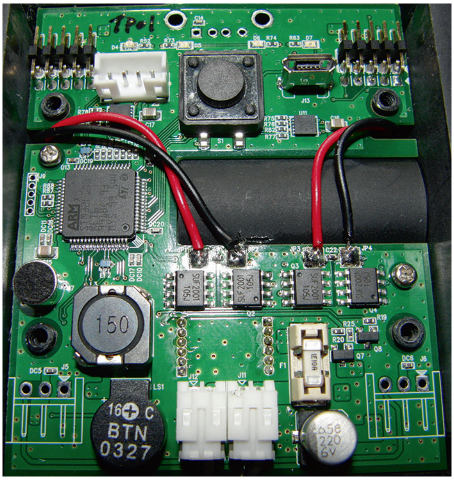


Fig. 2.21 Controller board for OLLOBOT

Register	Address	Size [byte]	Name	Description	Access	Default Value	Min Value	Max Value
0	0x0	2	Model Number	Model number	R	440	--	--
7	0x7	1	ID	DYNAMIXEL ID	R	200	--	--
79	0x4F	1	Green LED	Green LED on/off	RW	0	0	1
80	0x50	1	Blue LED	Blue LED on/off	RW	0	0	1
97	0x61	1	Input Voltage	Battery input voltage (unit: 0.1V)	R	--	0	255
112	0x70	1	Controller X-Axis Value	X-axis coordinate of joystick	RW	0	-100	100
113	0x71	1	Controller Y-Axis Value	Y-axis coordinate of joystick	RW	0	-100	100
128	0x80	1	Port 1 Servo Mode	Change left motor mode (wheel or joint)	RW	0	0	1
129	0x81	1	Port 2 Servo Mode	Change right motor mode (wheel or joint)	RW	0	0	1
136	0x88	2	Port 1 Motor Speed	Speed of left wheel	RW	0	-1024	1024
137	0x89	2	Port 2 Motor Speed	Speed of right wheel	RW	0	-1024	1024
156	0x9C	2	Port 1 Servo Position	Left wheel servo position	RW	--	-10240	10240
157	0x9D	2	Port 2 Servo Position	Right wheel servo position	RW	--	-10240	10240

Fig. 2.22 Control table for OLLOBOT controller (Courtesy of ROBOTIS)

will involve developing apps for Android devices (see Chap. 12 for more details). The various ROBOTIS SDKs for the OLLOBOT can be obtained here – http://www.robotis.us/ollobotsdk/.

2.5 System(s) Selection Criteria

When I first prepared the instructional materials for my "Embedded Robotics" course in early 2009, the only option was the CM-5 based systems: BIOLOID COMPREHENSIVE and BIOLOID EXPERT. But nowadays, as shown in the previous sections, we have many more choices for entry points into this "Robotics" journey. Furthermore, ROBOTIS has been putting in diligent efforts for maintaining backwards compatibilities for their software updates and their newly developed actuators and sensors so that the CM-5 based systems are still useful. For example, the following devices are compatible with the CM-5, although they may not function at their full capacities due to the 16 MHz clock speed on the Atmel AVR® chip:

- IR Array Sensor (2012) (http://support.ROBOTIS.com/en/product/auxdevice/sensor/ir_sensor_array.htm)
- Servo motor MX-28T (2011) (http://support.robotis.com/en/product/actuator/dynamixel/mx_series/mx-28at_ar.htm)
- Color video camera HaViMo2 (2010) (https://www.havisys.com/?page_id=8)

So let me go out on a limb and share with you these recommendations:

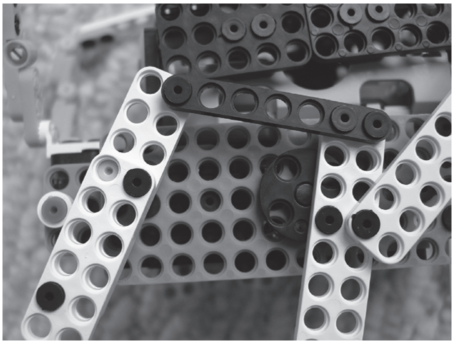
1. For the DIY user, the MiCom Training Kit can be an economical approach whereas the user can start with TASK and progress to ROBOTIS IDE. This kit could turn out to be a popular entry kit for a first look at the ROBOTIS robotics ecology.
2. The CM-530 STEM or PREMIUM kits would offer a fast MCU @ 72 MHz and the 5-pin GPIO connectors, and Embedded C is readily available (via WinARM and Eclipse). If you have little background in computer programming, you can get started on the RoboPlus Suite V.1 or V.2 and transition to Embedded C with WinARM or purchase additional CM-9.04-A's or B's for an Arduino™-style IDE, as all your hardware would still be compatible. ROBOTIS' e-Shop carries a Programming Guide showing how to use the RoboPlus V.1 software suite (http://www.robotis-shop-en.com/?act=shop_en.goods_view&GS=1486&GC=GD080400).
3. If you are already familiar with robotics and Arduino®, there is good reason to use the OpenCM9.04-C from the beginning. But you will have to purchase all the other components separately as you need them (http://www.robotis-shop-en.com/?act=shop_en.goods_list&GC=GD0803). The assembly instructions and parts lists for practically all ROBOTIS robots (except for the BIOLOID COMPREHENSIVE and PREMIUM series) are now available in the software tool called R + Design (http://support.robotis.com/en/software/roboplus2/r+design/r+design.htm) which essentially is a 3-D CAD tool (see Fig. 2.23). The OpenCM9.04-C is designed to work with the second generation of the RoboPlus software suite which has four tools: R+Design, R+Task, R+Motion and R+Manager. This option would afford the users the most flexibility in switching between ROBOTIS-style and Arduino-style user interfaces. If the user is further interested in interfacing the robot with a smart phone to access its facilities such as camera, audio-visual and messaging services, the OpenCM9.04-C can be programmed to work with the phone app R+m.PLAY700 which is available for Android and iOS systems.
4. When the SMART III system is available internationally, it would be a lower cost system for a freshman 1-credit course in introductory robotics, but it does not have room for expansion into more rigorous topics.

Fig. 2.23 R + Design tool

2.6 Review Questions for Chap. 2

1. What is the product brand name uniquely attributed to ROBOTIS Co.?
2. What was the name of the first commercial product from ROBOTIS?
3. Which are the robotic system(s) from ROBOTIS that use a rivet-like system for building robots?
4. Which ROBOTIS systems can be interfaced with SCRATCH 2?
5. Which ROBOTIS systems can be interfaced with R + MOTION?
6. Which ROBOTIS systems can be interfaced with a smart phone?
7. Can the phone app R + m.PLAY700 work with the DREAM system?
8. What is the name of the tool that displays the mechanical assemblies for most of ROBOTIS robots?
9. What are the micro-controllers currently supported by ROBOTIS?
10. List the robotic system(s) that are based on the STM32 family from STMicrolectronics Co.
11. List the robotic system(s) that are based on the ATmega AVR family from Atmel Co.
12. What is the name of the controller(s) associated with the open hardware-software initiative from ROBOTIS?

13. Which CM-XXX controller(s) can implement C-based computer programs from the user?
14. Which CM-XXX controller(s) run at 72 MHz?
15. List the four software tools that come with the RoboPlus Software Environment.
16. List three possible communication/interface protocols that are available to the user for use with ROBOTIS systems.
17. Which controller(s) support user-based firmware and boot loader?
18. Which controller(s) support an Arduino-based IDE?
19. Which controller(s) support an expansion board?
20. Name the four types of linkages that are found in a typical four-bar linkage system.
21. A __________ linkage is called a ___________ linkage when it can turn a full 360°.
22. **Identify** (i.e. draw) the correct links and **name** them properly for the four-bar linkage system as shown for the Cricket robot's front leg picture below.



http://www.springer.com/978-3-319-59830-7

Exploring Robotics with ROBOTIS Systems
Thai, C.N.
2017, XIII, 342 p. 381 illus., 364 illus. in color.,
Hardcover
ISBN: 978-3-319-59830-7