Contents

Chapter 1: Foundational Concepts

1.1 ROBOTIS DREAM II System Overview

ROBOTIS offers quite a large range of educational robotics kits to fit different user's ages and skill sets, please review this YouTube® video (https://youtu.be/Rxmgfose8wE) for an older (2017) survey of the ROBOTIS "family" of robots. It did not include the ENGINEER Kits 1 and 2 (https://www.youtube.com/watch?v=dJ0kP0mVPwE) which were released later in 2019-2020.

The DREAM II system (http://www.robotis.us/dream/) offers an extensive range of sensors such as Touch, Color, Magnetic, Passive IR and Temperature sensors. It can operate 2 different types of actuators (Continuous-Turn or Position-Control). Its Controller can be the CM-150 (https://emanual.robotis.com/docs/en/parts/controller/cm-150/) or CM-151 (https://emanual.robotis.com/docs/en/parts/controller/cm-151/) (see Fig. 1.1). Either Controller can be programmed with several software packages such as ROBOTIS' tools (R.TASK/R+BLOCK/R+PLAY700), SCRATCH (from MIT), or EDBOT (from Robots in Schools) for use with Python (see Thai, 2018 and Thai, 2019).

The present book was written to leverage special features of the Chromebook so that 3 Robotics Programming approaches could be presented to readers/users with different skill levels:

1) "Block Coding" with R+BLOCK for the youngest beginners.
2) "Menu-Guided Programming" with R+TASK for users with intermediate skills.
3) Standard Python Programming with THONNY IDE and Chromebook's LINUX Virtual Environment for more advanced users.

Fig. 1.1 Hardware Controllers for DREAM II kits: CM-151 (left) and CM-150 (right).

The DREAM II system shares into a common ROBOTIS System Design Paradigm which considers a typical ROBOTIS robot to be a Computer Network with four major components:

1) A Hardware Controller, CM-150/151 (see Fig. 1.1), is the “Main Brain” for the robot. It contains the user’s instructions for the robot to execute varied tasks as programmed by the user. Outwardly, there is no physical difference between CM-150 and CM-151, but the CM-151 has about 40% improvements in processing speed and sensor throughput. It also has an embedded BT-410 receiver with much improved wireless communications reliability (no packet losses for indoors usage per author’s experiences). The CM-150 uses an external BT-410 receiver via its UART port.

2) The Sensors component (such as NIR or Touch sensors – see Fig. 1.2), which helps the Hardware Controller get information about its surroundings. Sensor data are usually obtained as analog signals that need to be digitized by the Hardware Controller before the programmer can make use of the sensor inputs.

Fig. 1.2 Selected Sensors: NIR Sensor (left) and Touch Sensor (right).

3) The Actuators component (e.g., Servo Motors or LEDs – see Fig. 1.3), which allows the Hardware Controller to perform the appropriate robot actions upon the real world as programmed by the user. Advanced actuators such as servo motors have their own micro-controllers to work in concert with the Hardware Controller.

Fig. 1.3 Selected Actuators: Servo Motor (left) and LED Display (right)

4) A Remote Device, which can be an RC-100B module or a smartphone (see Fig. 1.4), or even a Windows PC or Chromebook via TASK's Virtual Controller. The Remote Device has a more flexible role depending on the specific device used and on its current role in this network of robotic components. For example, if the basic RC-100B is used with the DREAM system, it essentially functions as a NIR-based Sensor (attached to the user) informing the Hardware Controller about which "Up/Down/Left/Right" buttons had been pressed by the user. The basic RC-100B would require line-of-sight access between the physical RC-100B and the NIR receiver (IR-10) attached to the robot. When a smartphone (running the R+m.PLAY700 App) is used with the CM-150/151, it can act as a Bluetooth Sensor telling the CM-150/151 about which Touch Area on the phone screen had been pressed by the user, or it can act as an Actuator displaying appropriate graphics or videos as commanded by the CM-150/151 according to its programmed logic (see Thai (2019)).
When a Chromebook (CB) is used as a Remote Device, the issues become more complex. For example, if the TASK's Virtual Controller (see Fig. 1.5) is used, then the CB acts as a glorified Touch Screen informing the CM-150/151 about which virtual RC button had been pushed by the user. However, if the "LINUX+THONNY" tool chain is operating on the CB (see Chapter 4), then the CB takes on the role of a "Supervisory Controller" which is responsible for the main robot control logic, implemented via a Python program executed on the CB. While the CM-150/151 now acts only as a "simple" Hardware Controller handling wireless commands sent from/to the CB via the ROBOTIS communications protocol named "Remocon" (https://emanual.robotis.com/docs/en/parts/communication/rc-100/#communication-packet), and to pass them through to the robot's Actuators and Sensors, via a TASK program running on the CM-150/151. With this approach, the CB can even be configured to control multiple DREAM II robots.

Fig. 1.4 Selected Remote Devices: RC-100B (Left) and Smartphone (right).

Fig. 1.5 TASK's Virtual Controller Interface.

1.1.1 DREAM II - Levels 1-5 Edition

Since Spring 2018, the complete DREAM II System is available for the U.S.A. market and it includes 5 levels (http://www.robotis.us/dream/) which work in a cumulative scheme: i.e., users would have to purchase/start with Level 1, then go to Levels 2 through 5 consecutively:

1) Besides the various frame parts, the major components of Level 1 include 1 Low-Speed Geared Motor, 1 Micro-USB cable and 1 LiPo battery unit. Level 1 showcases 11 motorized but non-programmable robots.

2) Level 2 adds 1 CM-150/151 Controller, 2 Geared Motors, 2 Rubber Tires and more frame parts. Level 2 showcases 11 programmable robots; some are wheel-based, and some are linkage-based (i.e., "walking" robots) and all as autonomous behavior applications.

3) Level 3 adds 1 Servo Motor, 1 Remote Controller RC-100B and its NIR Receiver IR-10, and more frame parts. Level 3 showcases another 11 programmable robots using the Servo Motor and illustrating Remote Control techniques.

4) Level 4 adds 2 more Servo Motors, 2 External NIR Sensors, 2 Touch Sensors, 1 LED Module, and of course more frame parts. Level 4 showcases 12 additional robots with more sophisticated mechanical designs and behaviors, both autonomous and remote controlled.

5) Level 5 adds a set of Master and Slave BlueTooth modules to be used with the RC-100B, 1 more LiPo battery unit, and even more frame parts, including a Track system. Level 5 showcases 12 final wheel-based and track-based robots with autonomous and remote-controlled designs. Some applications use the more sophisticated Callback function which is a type of hardware-timer interrupt function.

1.1.2 DREAM II - School Set

The School Set is customized for the USA market and roughly corresponds to a bundle of the major components of Levels 1, 2 and 3 combined. Thus, both Autonomous-Behavior and Remote-Control applications can be achieved with the School Set (23 robot designs). This book is specifically written for the School Set (Fig. 1.6), but users need to also realize that extra components are available via the ROBOTIS e-Shop Network (https://www.robotis.us/ and https://www.robotis.com/).

Fig. 1.6 DREAM II School Set (Courtesy of ROBOTIS).

1.2 Sense-Think-Act Paradigm

Usually, the younger robotics students were surprised when I started my robotics short courses by mentioning that they had been using the Sense-Think-Act paradigm in everyday activities already. I used Fig. 1.7 to explain how humans use our Senses to let us know about the external World (i.e., Perception) and apply those Sensations into our Cognitive process (i.e., Thinking) to devise proper Actions/Reactions to the situation at hand. These Actions/Reactions in turn would change some aspects of the external World resulting in new Sensations triggering the next World-Human-Interactions cycle and so forth.

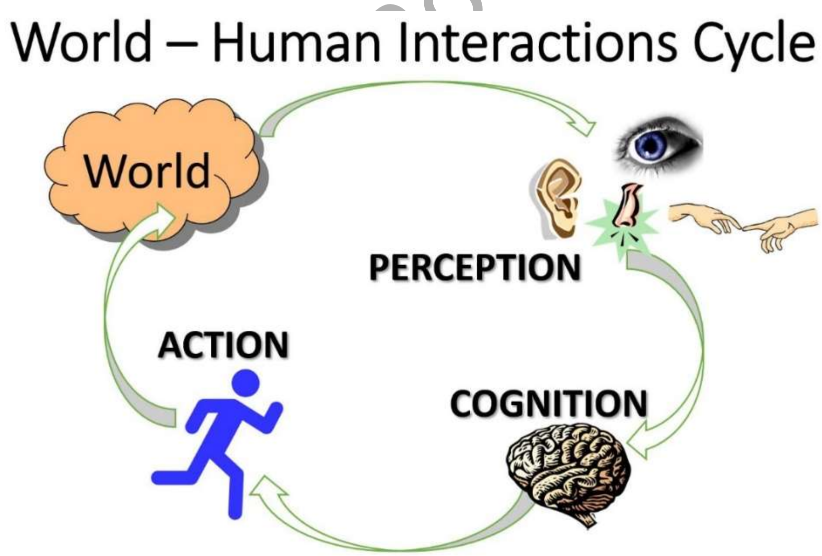


Fig. 1.7 World-Human Interactions Cycle.

Thus, in a way, doing Robotics is just trying to reproduce this interactions cycle onto a robot as shown in Fig. 1.8. Mechanical or electronic Sensors would convert their interactions with the World into Input Data to be sent to the Robot/Computer which was previously Programmed by the user to deal with Events deemed important to the operation of the Robot. Using this programmed Logic, the Robot Outputs appropriate commands to its Actuators which then could change the Sensors' perspective onto the World and therefore generate new Input Data for the Robot/Computer to process anew, through another interactions cycle. The robot can also be set into a Remote-Control mode, or an Autonomous-Behavior mode, or a mix between these two modes.

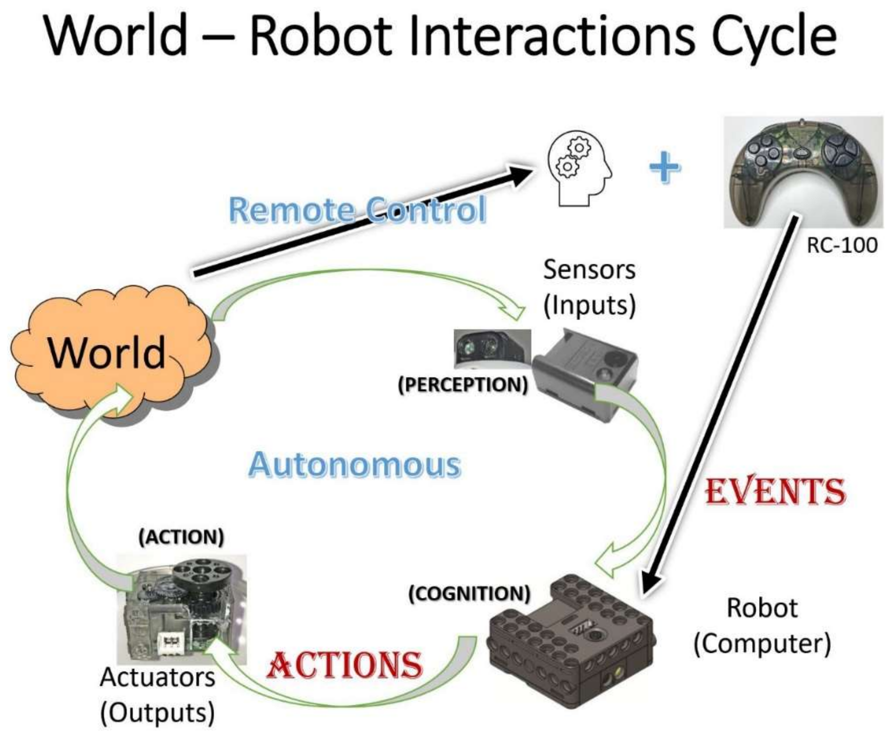


Fig. 1.8 World-Robot Interactions Cycle.

The Sense-Think-Act Paradigm will be revisited with more details in Chapters 2, 3 and 4 to illustrate its adaptations to specific hardware/software environments used for different robot systems.

1.3 Using DREAM II with Chromebooks

If the reader is using Chromebooks for the first time, the author is recommending this book "The Ultimate Chrome OS Guide" by Keith Meyers (2021-2022). It is available in Kindle format at this link https://www.amazon.com/s?k=ultimate+chrome+os+guide&crid=17COI4WVHVCE, and it has different versions depending on the actual Chromebook model that the reader already possessed or plans to purchase. The author used three different Chromebook brands in developing DREAM II projects for this book (see Appendix A for discussions about a "median" Chromebook to use for materials presented in this book).

Currently (April 2022), all ROBOTIS software tools such as BLOCK, TASK V2/3 and other mobile Apps were developed for devices running on Android OS, so their uses on Chromebooks are still "quirky". Appendix A lists possible use issues of the DREAM II Hardware and Software suite and the author's "work-arounds" for those issues. **The author recommends readers to read Appendix A before continuing with Chapters 2, 3 and 4.**

Out of its shipping box, the DREAM II School Set is fully functional on a Windows PC platform equipped with a USB port using the provided micro-USB cable, but this option is not workable at present on Chromebooks, as USB devices are not yet fully supported in Chrome OS. This means that Chromebook users will need to buy extra BT communications devices from ROBOTIS to use programming tools such as R+BLOCK, R+TASK or Python:

- R+BLOCK and R+TASK work with either BT-210 (https://www.robotis.us/bt-210/) or BT-410 (https://www.robotis.us/bt-410/) receivers connected to the CM-150/151 UART Port.
- On the Chromebook's side, procedures for BT connections are "quirky":
    - The BT-210 can be paired via "Settings" in Chrome OS, but **not** for the BT-410.
    - But inside R+BLOCK and R+TASK V2 and V3, both types of BT receivers can be scanned for, paired, and be used in programming the CM-150/151.
    - Additionally, when programming in Python via the Linux Virtual Environment, both BT-210 and BT-410 receivers are not recognized by the Linux OS. But if a BT-410 USB dongle is used on the Chromebook's side in conjunction with a BT-410 receiver on the CM-150/151 side, then THONNY IDE would work fine with the DREAM II system (https://www.youtube.com/watch?v=FdRiB44DvDk).

In summary, for programming DREAM II robots with Chromebooks, the author recommends readers/users to purchase a set of BT-410 USB dongle and BT-410 UART receiver in case of a CM-150, and only a USB BT-410 dongle for the CM-151 as it already has an embedded BT-410 receiver (see Fig. 1.1).

It is rumored that a newer USB BT Dongle will be available to work with the CM-151 to take advantage of its higher baud rate (1 Mbps) by the end of 2022.

Additional ROBOTIS modules are needed for projects described in this book and they are listed below:

- Two IRSS-10 sensors (https://www.robotis.us/ir-sensor-irss-10/).
- One LED Module LM-10 (https://www.robotis.us/led-module-lm-10/).

This web site (https://emanual.robotis.com/docs/en/software/#r-20) contains download links for all needed DREAM II software tools. In Chapter 2, we will need to use the R+m.Design App which can be installed from this web link https://play.google.com/store/apps/details?id=com.robotis.mdesign.

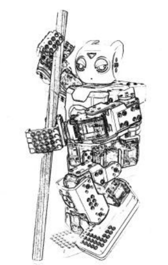
1.4 References

Myers KI (2021) The Ultimate Chrome OS Guide: Acer Chromebook 315.

Thai CN (2017) Exploring Robotics with ROBOTIS Systems (2nd Edition). Springer Publishers International, New York.

Thai CN (2018) Learning Robotics with ROBOTIS DREAM System. CNT Robotics LLC Publisher. Printed by Amazon KDP.

Thai CN (2019) Advanced Application Programming with ROBOTIS DREAM System. CNT Robotics LLC Publisher. Printed by Amazon KDP.



Chapter 2: Using R+ Block

Block Programming/Coding tools are well suited for the youngest beginners (5+) in computer programming, in general, and in robotics programming, in particular (https://childhood101.com/block-coding-websites-kids/, https://www.shaperobotics.com/block-based-programming/).

2.1 R+SCRATCH and R+BLOCK

For the CM-150/151, ROBOTIS offers two tools to work with Block Programming/Coding:

- R+SCRATCH is a helper application that runs on Windows PCs only and is designed to work with the MIT SCRATCH 2 Offline Editor (https://emanual.robotis.com/docs/en/software/rplus2/scratch/, https://scratch.mit.edu/download/scratch2). If the reader is interested in using this tool, please refer to Thai (2018). Currently, R+SCRATCH works with CM-150 only.
- R+BLOCK is based on Google Blockly and is originally designed for Android Mobile Devices (https://developers.google.com/blockly, https://emanual.robotis.com/docs/en/software/rplus2/rplus2_block/). For a comparison between Blockly and Scratch, the interested reader can visit this link https://www.robotlab.com/blog/blockly-vs-scratch-whats-best-for-me).

The R+BLOCK App was originally designed for Android Mobile Devices, but it is also usable on Chromebooks (https://play.google.com/store/apps/details?id=com.robotis.rblock). However, some issues remain and are discussed/resolved in Appendix A and later in this Chapter. True to its namesake, R+BLOCK uses a BLOCK-based interface to ease interaction and learning for the youngest beginners in robotics programming/coding (see Fig. 2.1). **Please install this App on your Chromebook if you have not yet done so**.

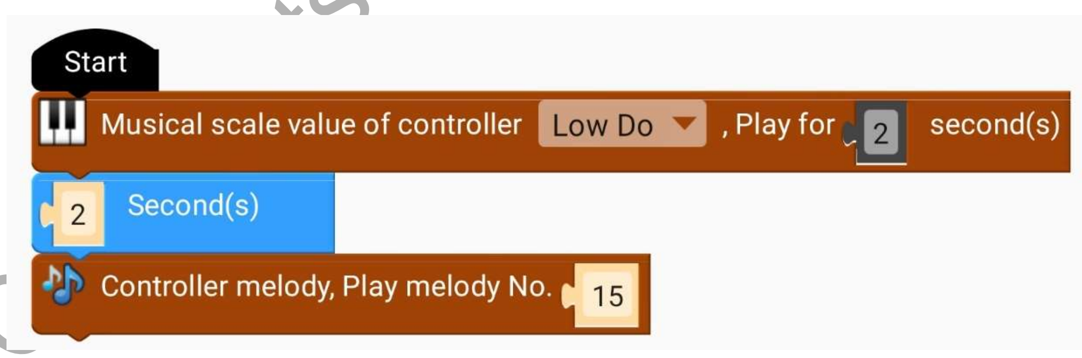


Fig. 2.1 A Basic R+BLOCK program/code.

This ROBOTIS web site contains good information for using the R+BLOCK App from software installation to how to navigate and how to use its Menu System (https://emanual.robotis.com/docs/en/software/rplus2/rplus2_block/). The current version of R+BLOCK is V.1.1.3 (c. 2020). **The author recommends that readers go over these materials and practice on their own Chromebooks to get some familiarity with the Menu system of R+BLOCK**. Readers can also watch this Video Tutorial for a "minimal" How-To Primer about R+BLOCK on Chromebooks (https://youtu.be/Fn-j2-X602Q).

Currently (Spring 2022), R+BLOCK works well with the CM-150 on Chromebooks, and only partially with the CM-151 as this is a newer controller, however this issue will likely be taken care of by ROBOTIS in the near future.

Chapter 2 is written for a complete beginner in robotics and computer programming, and we will be using the "Basic Avoider" platform and its variance (Fig. 2.2) to learn how to use the R+BLOCK App. A "Spiral" learning approach (Bergin, 2012) will be used for instructional purposes where the author would show how to use the R+BLOCK tool with projects having increasing scope and difficulty.

2.2 Construction of Avoider/Follower Robot

First, let's build a Basic Avoider/Follower robot as the demonstration platform (see Fig. 2.2) for various software projects presented in upcoming sections of this Chapter 2. Please watch Video at https://www.youtube.com/watch?v=TORJnsplPYI about how to use the R+DESIGN App to construct this robot, but only up to **Step 201 of the DREAM L2 Avoider design**. The final steps are to add two IRSS-10 sensors to the Back Side of this robot and to Port 3 and Port 4 of the CM-150/151 (also see Fig. 1.1 for a view of the finished robots).

These IRSS-10 sensors (https://www.robotis.us/ir-sensor-irss-10/) will have to be purchased separately as they do not come with the DREAM II School Set. Technical details for the IRSS-10 Sensor are available at https://emanual.robotis.com/docs/en/parts/sensor/irss-10/.

Fig. 2.2 Basic Avoider demonstration platform.

We are now ready to learn how to use the R+BLOCK App and most importantly how to learn basic Robotics Programming Concepts and Techniques using this Avoider platform. Please keep this web link (https://emanual.robotis.com/docs/en/software/rplus2/rplus2_block/) open for quick references when needed.

2.3 R+BLOCK Menu System

Fig. 2.3 depicts the Start-Up Window for R+BLOCK in Chrome OS:

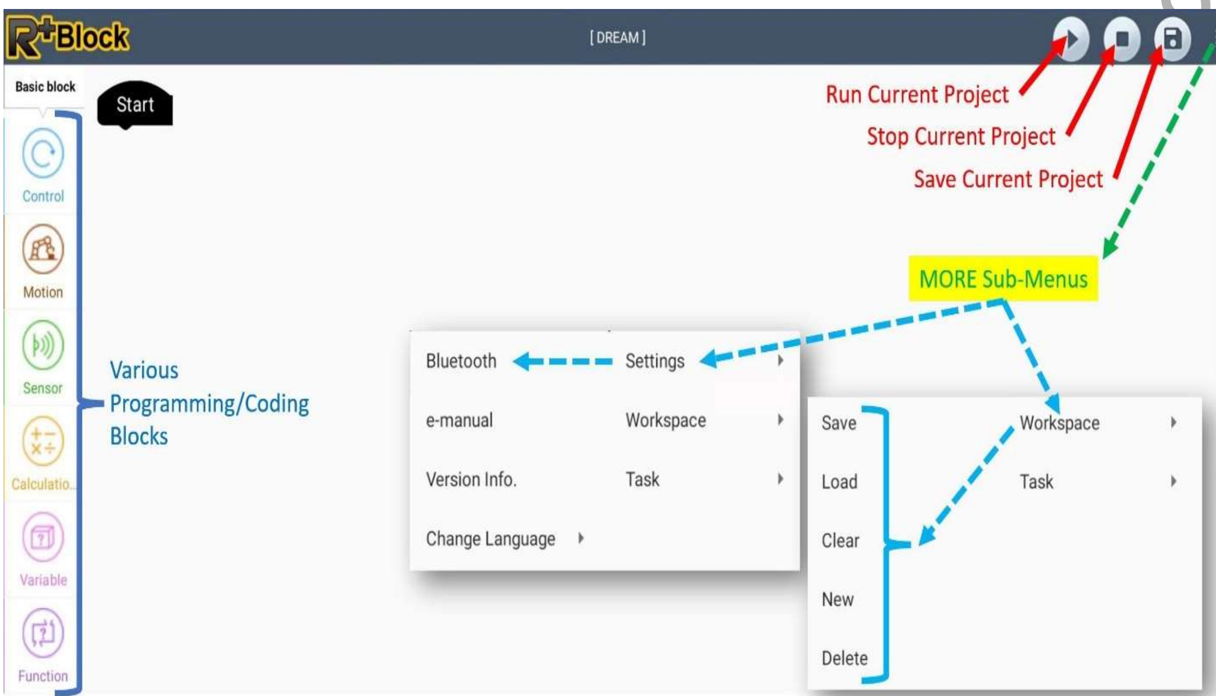


Fig. 2.3 R+BLOCK's Start-Up Window.

- On the Left Edge, ones can see the various types of Programming/Coding Blocks that can be used in a R+BLOCK Project: **Control, Motion, Sensor, Calculation, Variable** and **Function**.
- On the Top Right corner, ones can see 3 Buttons involving the Current BLOCK Project: **Play, Stop** and **Save**, and the **More** Button (3 vertical dots).
- The "**More**" Button will allow the user to access 3 more sub-menus: **Settings, Workspace** and **Task**. With **Settings** chosen and drilling down to **Bluetooth**, the user can scan for the BT-210/410 receivers connected to the CM-150/151. With **Workspace** chosen, the user can select various actions related to Projects (*.rpb files): **Save, Load, Clear, New** and **Delete** (see Appendix A for the exact location of the saved project files, i.e., the **DREAM** Folder).
- The **Task** sub-menu allows the user to download the Current BLOCK Project directly into the CM-150/151 as a TASK binary code, or to convert it into a TASK V.2 format/file (*.tskx) saved to the same **DREAM** Folder as for BLOCK Projects. This option will be discussed in more details within a later sub-section.

If the user selects **More** >> **Settings** >> **Bluetooth**, then the Bluetooth Scan Panel would pop up (see Fig. 2.4):

- Currently (4/2022), the information displayed vis-à-vis BT-410 and pre-pairing of BT-210 at the Chrome OS Settings level is **incorrect**, because the R+BLOCK's internal BT scanning tool works fine for BT-210/410 receivers used on a CM-150 and even for the embedded BT-410 found on a CM-151.

Fig. 2.4 R+BLOCK's Pop-Up BT Scan Panel.

- However, if the reader uses a BT-410 USB Dongle in combination with a BT-410 receiver on the robot, this internal BT scanning tool will fail (because R+BLOCK and Chrome OS do not support this type of USB device yet). The combo BT-410 Dongle+Receiver only works properly with the Linux Virtual Environment (i.e., when we do Python Programming in Chapter 4).

Fig. 2.5 shows the actual Programming/Coding Blocks that are of the **Control** type (Left Panel), and from the top they are:

- TIME DELAY Block for user-set number of seconds.
- ENDLESS LOOP as a Forever Repeat Do Loop.
- COUNTER-BASED Do Loop.
- CONDITION-BASED Do Loop.
- Simple IF structure.