

RHODES UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

HONOURS PROJECT: LITERATURE REVIEW

Integrating real-time image processing into rapid prototype robotics

Principal Investigator:

JEAN-CLIVE BAILEY

Supervisor:

Mr. JAMES CONNAN

Friday 29th May, 2015

1 Introduction

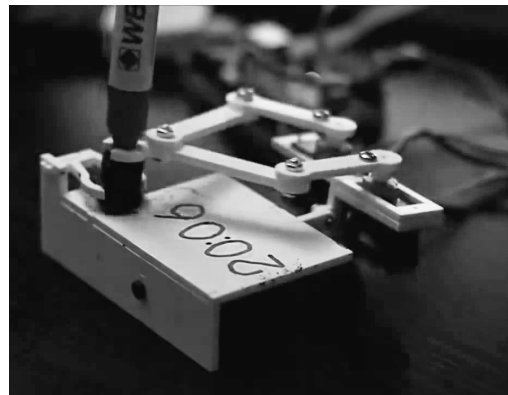
This document will review research relevant to the use of small scale robotics within the computer science discipline, the methodologies used to build robotic units, and how the techniques were incorporated in their respective scenarios. Since the goal of this project is to evaluate how robotics systems succeed at solving a goal driven task, specific attention is given to research documentation that focuses on the effectiveness of project development in the areas of LEGO Mindstorms NXT and embedded Raspberry Pi systems. Additionally the focus of this project, the Plot Clock, will be kept in close consideration.

2 The Plot Clock

The Plot Clock is an open-source hobby kit designed by Johannes Heberlein and published online¹. The entire kit, which includes .ino source code and Computer Aided Design (CAD) templates, is free to download and can be built using an Arduino Uno, three servos, and a whiteboard marker[1]. The Arduino is programmed in C. The purpose of the clock is to have a mechanical drawing arm pick up the board marker and use to plot the current time on an drawing surface and erase it with a wiping cloth. This robotic system makes use of a loop to maneuver the arm to each digit position in the clock and plot it accordingly. An example of the Plot Clock can be seen in a demonstration video². Two frames from the video can be seen in Figure 1, which demonstrates the Plot Clock's primary function.



(a) Start of writing



(b) End of writing

Figure 1: Heberlein's Plot Clock in action

¹<http://www.thingiverse.com/>

²<https://www.youtube.com/watch?v=NiGG9Lcrni8>

2.1 Proposed evolution

An important observation to note is that the Plot Clock does not incorporate any other form of logical operation to react with its environment. The non-deterministic characteristics of the system allow for the construction of a bespoke solution that is tailored specifically for the problem space only. The Plot Clock design comprises of a simple non-flexible software component and a re-usable mechanical drawing arm. Thus this project aims to continue the works identified, but will not re-invent the plotting robot. Instead a pre-designed robot will be identified based on its merits and depending on the resources available.

An initial proof of concept for the robot (physical robot and software agent) requires a firm analysis of the resources provided by Heberlein. To ascertain the mechanical functionality shown by the Plot Clock, a studious inspection of the robot's servo movements is required. This can be achieved through the hobby kit documentation and prototype experimentation. An immediate key concern is the degrees of freedom the drawing arm has – it is questionable whether the robot needs the capability to draw in arc when a square font similar to a seven-segment display may be sufficient for plotting purposes.

2.2 Work-flow

A plot-cycle is the action of the robot plotting one complete scenario – in this instance the scenario is the current time. The work-flow of the Plot Clock outlines how the robot behaves during one full plot-cycle. To assist in the understanding of how the clock fundamentally works, it can be deconstructed to the following work-flow:

- Record the current system time and adjust with known duration of writing offset.
- Segment each digit of a four digit clock (HH:MM) into plotted points.
- Move writing arm to digit origin position.
- Using the fixed origin, draw the numeric along a designated route.
- Repeat for each digit until the clock is written out.
- Move pen back to rest position.
- Move pen across entire board without lifting it, thereby erasing the board.

From this observation, three fundamental states in which the Plot Clock operates are outlined:

- **Sleeping** - The robot is waiting for the next work-flow.
- **Plotting** - The robot is currently executing a work-flow.
- **Initializing** - The robot is setting up initial parameters.

The prototype Plot Clock must at minimum adhere to this specification to be deemed a suitable reproduction. The absolute baseline for this project is the ability to write the clock and be capable of applicable improvement.

2.3 Servo analysis

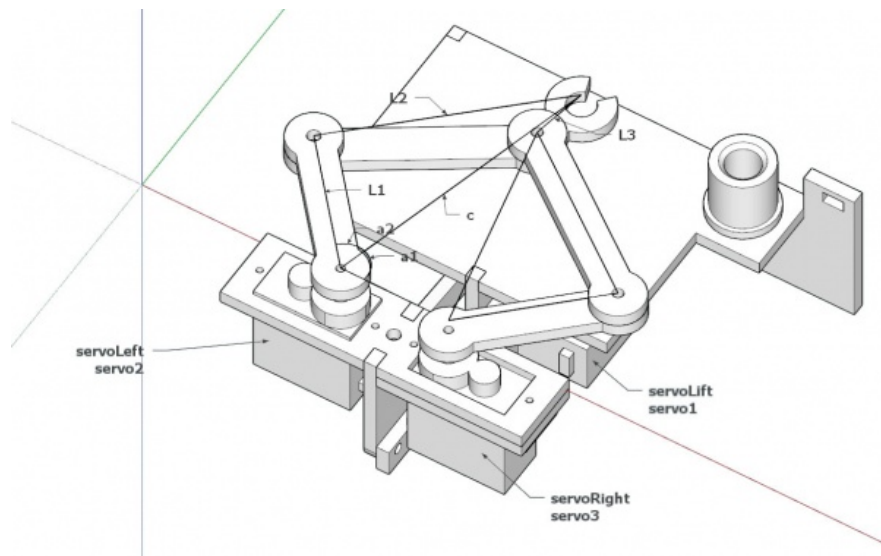


Figure 2: Plot Clock isometric design[2]

The Plot Clock has three primary servos as shown in Figure 2 which operate the movement of the marker. An analysis of Heberlein's source code, `plotclock.ino`, shows that the robot uses a collation of switch cases to maneuver the marker into the various digit origin and plotting points. A sample of one such case can be seen below in Listing 1.

Listing 1: An example servo movement to draw the numeric '0' in `plotclock.ino`[3].

```

1  case 0:
2    drawTo(bx + 12 * scale, by + 6 * scale);
3    lift(0);
4    bogenGZS(bx + 7 * scale, by + 10 * scale, 10 * scale, -0.8, 6.7, 0.5);
5    lift(1);
6    break;

```

2.4 The mathematics behind the code

A further analysis of the source code reveals the mathematical calculation which the Arduino board executes to control the arm movement. Figure 2 shows the lengths at which the arms operate and Listing 2 shows how the plot points are calculated. Note that the servo functions in lines 16 and 31 are called from `Servo.h` – a standard library that allows the Arduino board to control RC (hobby) servo motors [4]. This library is available as a download for the Arduino native Integrated Development Environment (IDE).

Listing 2: The point to point control function in `plotclock.ino`[3].

```
1
2 void set_XY(double Tx, double Ty)
3 {
4   delay(1);
5   double dx, dy, c, a1, a2, Hx, Hy;
6   // calculate triangle between pen, servoLeft and arm joint
7   // cartesian dx/dy
8   dx = Tx - O1X;
9   dy = Ty - O1Y;
10
11  // polar length (c) and angle (a1)
12  c = sqrt(dx * dx + dy * dy); //
13  a1 = atan2(dy, dx); //
14  a2 = return_angle(L1, L2, c);
15
16  servo2.writeMicroseconds(floor(((a2 + a1 - M_PI) * SERVOFAKTOR) + SERVOLEFTNULL));
17
18  // calculate joinr arm point for triangle of the right servo arm
19  a2 = return_angle(L2, L1, c);
20  Hx = Tx + L3 * cos((a1 - a2 + 0.637) + M_PI); //36,5
21  Hy = Ty + L3 * sin((a1 - a2 + 0.637) + M_PI);
22
23  // calculate triangle between pen joint, servoRight and arm joint
24  dx = Hx - O2X;
25  dy = Hy - O2Y;
26
27  c = sqrt(dx * dx + dy * dy);
28  a1 = atan2(dy, dx);
29  a2 = return_angle(L1, (L2 - L3), c);
30
31  servo3.writeMicroseconds(floor(((a1 - a2) * SERVOFAKTOR) + SERVORIGHTNULL));
32 }
```

3 Lego Mindstorms NXT

The LEGO Mindstorms systems, a joint venture between the LEGO® Corporation and the Massachusetts Institute of Technology (MIT), is a series of programmable robotic kits predominantly utilized as a teaching tool in introductory programming courses. The second generation released in June 2006 and is called the NXT series, which replaced the first series called Robotic Command eXplorers (RCX) [5]. The Mindstorms Kit is a popular choice among robotics enthusiasts and academia since it provides desirable properties as opposed to competing units like Adept MobileRobots's Pioneer robots [6] and MIT's HandyBoard and Super Cricket [7].

Klassner and Anderson [8] demonstrated that LEGO Mindstorms became a suitable platform for college students to investigate a broad range of topics and they had specifically chosen the LEGO Mindstorms for a few distinct reasons – three are applicable to this project:

1. **Cost:** A single Mindstorms kit, with 750 construction pieces, sensors, and programmable hardware, costed approximately \$200. At the time this was a quarter of the HandyBoard-based robot kit and a mere tenth of the MobileRobot kit.
2. **Flexibility:** The Mindstorms platform supports a suite of reusable snap-together sensors, effectors, building blocks and control unit that can serve as a basis for a variety of projects.
3. **Professional Curiosity:** Other liberal-arts college instructors cited that the Mindstorms kit was a great source for robot chassis construction.

It is worth noting that Rhodes University had granted the use of a LEGO Mindstorms Education NXT kits to the development of this project – this significantly speeds up the prototyping phase since the kit was made available early within the project life-cycle. Additionally, the source material for most of this research was based on the first generation RCX systems. Moreover, the technology has improved with the latest iteration of the Mindstorm series; the NXT is regarded a major improvement over the RCX platform [9, 10].

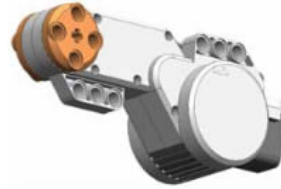
3.1 Technical Specification

The LEGO Mindstoms Education NXT kit is composed of a software component and several hardware components, mostly parts that will not be needed for this project. The technical specification [11] of applicable components are as follows:

3.1.1 Hardware



(a) NXT Intelligent Brick



(b) Interactive Servo Motor

Figure 3: The key Mindstorms NXT components

Intelligent NXT Brick

- 32-bit ARM7 microprocessor.
- 256 Kbytes FLASH, 64 Kbytes RAM.
- Unlimited programming slots.
- Bluetooth wireless communication.
- USB 2.0 port (slave only).
- 4 input ports, 6 wire digital platform.
- 3 output ports, 6 wire digital platform.
- Programmable Dot Matrix Display 60 x 100 pixels.
- Loudspeaker 8 KHz sound quality.
- Many LEGO Technic holes allow for easy and fast building.

Interactive Servo Motor x 3

- Built-in rotation sensor.
- Measures exact speed and distance which reports back to the NXT.
- Many LEGO Technic holes allow for easy and fast building.
- Allows for precise motor control with an accuracy of 1 degree step control.

3.1.2 Software

Lego Mindstorms NXT 2.0 Software

- Powerful, easy-to-use software designed to work with the NXT brick.
- Teaches users the fundamentals of NXT-G
- Powered by LabVIEW .
- Icon based, user friendly programming tool.
- Progressive platform for students from primary school to university level.
- Windows and OS X compatible.

3.2 NXT Brick Evaluation

Wan [10] explains that the NXT brick contains an ARM Micro-controller which consists of a microprocessor core and programmable peripheral control logic. The stock user software develops code using NXT-G, the LEGO programming language, and is compiled into a byte-code-based virtual machine instruction set, termed NXT byte-codes. This is executed on the NXT Virtual Machine (NXTVM) which resides as firmware on the NXT brick. This architecture is similar to that of Java byte-codes, which is a visualized instruction set that executes on a virtual processor using a Java Virtual Machine. Due to this concession it is possible to program the robot with an alternative programming language, provided that the compiler can compile into low level byte-code which the NXTVM is able to interpret.

In reality, this is the preferred method of programming the NXT (and RCX) bricks; the NXT-G environment interface is slow, lacks programming fundamentals like events and floating points, and is widely considered to be used exclusively as a teaching tool [12, 13, 9, 14]. There is a remarkable interest in the robotics development community to build supportive programming environments, and several independent Open Source Software (OSS) languages are recognized in academia as suitable alternatives [15]. The LEGO corporation is also very supportive of developers, and even released the firmware source code for the latest Mindstorms EV3 series [16]. Thus, further endorsing the use of third-party languages and software like JAVA and RobotC [17].

3.3 Alternatives to NXT-G

3.3.1 RobotC

ROBOTC [18], frequently spelled as RobotC, is a full commercial IDE used to program and control LEGO EV3, NXT, RCX and Arduino robots using a language based on the ANSI C programming language . The IDE features: debugging support, user tutorials, an intelligent compiler to help correct syntax errors, and interfacing with gaming controllers via a computer to the NXT. The programming fundamentals are identical to C and it supports most C structures such as pragmas, coding blocks and functions. The software is neither open source nor free; an individual 365-day single-user license costs \$49.00 [19].

3.3.2 GNAT GPL for Lego Mindstorms - Ravenscar Edition

GNAT GPL for Lego Mindstorms - Ravenscar Edition [20] is a GPL port for the GNAT compilation systems to the NXT platform. The language is an education-oriented project originally developed at MIT Media Lab, but has since evolved into a successful commercial project used in multiple universities and schools globally. The Ravenscar Edition supports Ada-level real-time and concurrency features and, as of the 2010 release, does not rely on any operating system. The software package includes support for Ada 2005, SPARK and RavenSPARK, high-level drivers for the NXT brick, sensors, motors and Bluetooth, written in Ada. The software is free to use, but is not OSS.

3.3.3 NeXT Byte Code

NeXT Byte Code (NBC) [21] is syntactically an similar to Assembly but can be used to program the NXT. The NBC compiler translates source code into NXT byte-codes and can be executed directly onto the NXT brick. Although the format and preprocessor is similar to Assembly, the language is not general purpose and many restrictions are inherited from the LEGO byte-code interpreter. NBC supports a full Application Programming Interface (API) that describes system functions, constants and macros are that available to programs – the API is defined similarly to a C-style header file which is automatically included in every NBC program. NBC is completely free and released under the Mozilla Public License [22].

3.3.4 Not Quite C and Not eXactly C

Not Quite C (NQC) [12], and its NXT derivative Not eXactly C, are programming languages specifically designed to interface with LEGO robotics . As the names imply, NQC and NXC

are based on ANSI C programming language and share similarities with 3.3.1, but lack the developed IDE. Both languages are supported by the Bricx Command Center, a text-processor suite that also allows commands to be sent to the NXT brick and other sundry functionalities. These languages underwent extensive development since their conceptualization by Overmars in 1999, and multiple revisions by Hansen between 2002–2011 [23].

3.3.5 leJOS

LeJOS [24] is a firmware replacement for the Minstormds programmable bricks. It currently supports the RCX bricks and leJOS NXJ supports the NXT bricks. The package includes a JVM, allowing the Java programming language to be used to control the robot. The LeJOS NXJ provides support for access to the robot’s input and output ports, allowing access to the standard sensors and motors – this further supported with Java APIs that control these systems. By taking advantage of the object-oriented structure of Java, LeJOS NXJ programmers are able to work with high-level abstractions without a concern for low-level hardware access handlers. The entire LeJOS project builds on Java platform and has seen two plugins for leading IDEs: Eclipse and Netbeans.

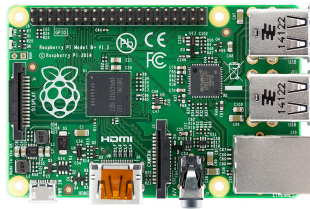
3.4 Scope of LEGO Mindstorms

Robots can vary from simple remote control responding peripherals to autonomous machines that are able to sense and ultimately responds to environment with the inclusion of Artificial Intelligence. A true independent robot is beyond the scope of this project but could prove to be a unique challenge for future Computer Science students to attempt. The LEGO Mindstorms systems have definitely proven to be robust systems and is constantly expanding with various community-led innovations.

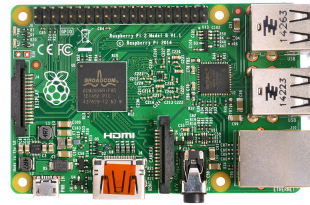
A prime example is the Pyro, a robotics programming environment designed to allow users to delve further into sub systems at a low-level and across multiple robot platforms[25]. Blank, Meeden and Kumar originally designed Pyro to be a simple robotics API plugin as a set of classes and libraries to explore advanced robotics issues on different platforms, but the system was fast enough to be used in their main research. This exemplary use of platform development further establishes the use of LEGO Mindstorms NXT as a cornerstone in this project.

4 Raspberry Pi Robotics

The Raspberry Pi (RPi) is a low cost, credit-card sized computer with multiple peripheral ports that allow connection to traditional Human Interaction Components like a mouse, keyboard and monitor, but it additionally hosts a General Purpose Input Output connector that interfaces with electronic circuits [26] – the quintessential piece for building a Raspberry Pi based robot. Over the last five years the Raspberry Pi has been released as four versions, each building upon the strengths of the previous generation and providing a multitude of features which will be explored in Table 1.



(a) Generation 1 Model B+



(b) Generation 2 Model B

Figure 4: The Raspberry Pi units

4.1 Technical Specification

4.1.1 Hardware

Raspberry Pi	Model A+	Model B	Model B+	2 Model B
Price	\$19.99	\$39.99	\$29.99	\$39.99
Chip	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
Processor	ARMv6 single core	ARMv6 single core	ARMv6 single core	ARMv7 quad core
Processor Speed	700 MHz	700 MHz	700 MHz	900 MHz
Voltage and Power Draw	600mA @ 5V	600mA @ 5V	600mA @ 5V	650mA @ 5V
GPU	Dual Core VideoCore IV	Dual Core VideoCore IV	Dual Core VideoCore IV	Dual Core VideoCore IV
Size	65x56mm	85x56mm	85x56mm	85x56mm
Memory	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz	1 GB SDRAM @ 400 MHz
Storage	Micro SD Card	SD Card	Micro SD Card	Micro SD Card
GPIO Pins	40	26	40	40
Ethernet	None	10/100mb Ethernet RJ45 Jack	10/100mb Ethernet RJ45 Jack	10/100mb Ethernet RJ45 Jack

Table 1: Raspberry Pi comparisons [27]

Additionally, all the Raspberry Pi models share the following features:

- **Audio:** Multi-Channel HD Audio over HDMI, Analog Stereo from 3.5mm Headphone Jack
- **Video Output:** HDMI Composite RCA
- **Supported Resolutions:** 640x350 to 1920x1200, including 1080p, PAL & NTSC standards

- **Power Source:** Micro USB

4.1.2 GPIO

The GPIO port on the Raspberry Pi is a 26x or 40x male header connector on the edge of the board. The GPIO does not support direct interaction to peripherals like servo motors, but can interface to a circuit board or breadboard [26]. The GPIO pins are versatile and many of its characteristics can be modified from software – though the properties are set for the entire GPIO block and not on a pin-by-pin basis [28]. When configured as an output, the GPIO simulates the Raspberry Pi as a switch and battery source; the pins can provide 3.3 volts (3v3) on High and 0 volts (0v) on Low [29]. OSS driver software has been developed for C and Python - making the Raspberry Pi an ideal choice for the Plot Clock.

There are some risks to working with integrated circuits and the GPIO ports; the pins connect directly to the processor and do not have any protection to protect the processor against faulty wiring or high voltage on the pins. An example fault of 5V on the 3v3 pins could permanently damage the Raspberry Pi.

4.2 Software

The Raspberry Pi primarily uses Linux-kernel-based operating systems [30]. The install manager for the Raspberry Pi is called NOOBS. The operating systems included with NOOBS are:

- Archlinux ARM
- OpenELEC
- Pidora (Fedora Remix)
- Puppy Linux
- Raspbmc and the XBMC open source digital media center
- RISC OS, the operating system of the first ARM-based computer.
- Raspbian (recommended for Raspberry Pi 1)
- (Raspberry Pi 2) Windows 10 - Microsoft announced February 2015, it will offer a free version of the to-be-released Windows 10 running natively on the Raspberry Pi. [31]

4.2.1 Recommended Operating system

The documentation found on Raspberry Pi units used for robotics (Pibots) appear to be in favor of Raspbian [32] since it is easy to understand and can be developed intrinsically [26]. The majority of Pibots are more designed around the language used to control them, as mentioned in 4.1.2, the RPi.GPIO library contributes towards deciding factors for what software package to use. At this point in time it is unclear which software is ideal, further experimentation and investigation will need to be made.

4.3 Raspberry Pi used in academia

4.3.1 RPiCluster

The Raspberry Pi Cluster project [33], an initiative by Joshua Kiepert, was to construct a RPi-based 32-node Beowulf cluster: a collection of identical, commodity computer hardware-based systems that are networked together and run a variation of parallel processing software that allows each node in the cluster to share data and computation. Kiepert required a number of computers in a common network as a platform for his PhD thesis on the development of a novel data-sharing system for wireless sensor networks. The motivation behind creating his own cluster as opposed to using the Boise State University Computer Science Department's Onyx Lab, a collation of 32 quad-core processor Intel computers, was to enable Kiepert complete freedom in the availability of the cluster at any time and concise control over the node software environments.

The reasoning behind the RPi units for his project was because the units provided the option of continued development of Message Passing Interface projects, the units have the unique low level hardware interfaces for embedded systems (via the GPIO) and the total project cost was relatively inexpensive to develop. Kiepert concluded that the RPiCluster proved successful and since its completion he was able to move his dissertation work exclusively onto it – Kiepert also stated the system performance suited his simulation needs perfectly [34].

4.3.2 mPowerProfile

The mPowerProfile [35] was an environment designed to measure the automated power measurement of mobile computing platforms. The motivation for the project was to ascertain the run-time power measurements on mobile platforms; this information would be used in studies that aim at developing analytical models for energy estimations derived from real

world problems. The platform relied on minimal invasive instrumentation of a mobile platform using a shunt acquisition system (DAQ) for sampling voltage at the shunt resistor. The mPowerProfile software ran on a developed workstation and interfaced to the mobile platforms via serial connections and to the DAQ via USB connection.

Milosevic *et. al.* used both the Texas Instruments Pandaboard and a Raspberry Pi in their study. The case studies took interest in the minimizing storage capacity requirements and energy costs of data communication as these properties were critical aspects in all mobile platforms. The study demonstrated the use of mPowerProfile in evaluating the energy efficiency of uncompressed and compressed data transfers from the Pandaboard and RPi to a remote server and vice versa. The RPi was specifically chosen since it was relatively new in the mobile platform market and was an affordable comparison against the Pandaboard. The study concluded with the energy overhead findings and Milosevic *et. al.* evaluated the mPowerProfile software performance on both platforms.

4.3.3 Remote control RPi Smart Projectors

Chheda *et. al.* [36] put forth the idea of enhancing the quality of service of technology-aided teaching through the use of a RPi and its web interface to store presentation files which would then be remotely accessed via a smart phone and be displayed via projector. The RPi was selected for this project due to availability of an HDMI port and IEEE 802.11g Ethernet slot – both features facilitated the display of content and the upload of files respectively. Furthermore, the RPi was the selected candidate due to its cost effectiveness, a mere tenth of a standard laptop price, and energy efficiency since it consumes only 5.0V of power.

The RPi would have multiple software packages like Apache, PHP and MySQL installed and, when configured, would display a web-interface using HTML-CSS-PHP technologies to display all local content stored on the device. Finally, the system utilized a NetIO multi platform smart phone application as the remote control element; this package required some configuration but allowed most smart devices to interface with the RPi. The project concluded that the proposed device offered the same functionality as a laptop and only required the use of a smart phone which was readily available within their system environment scope.

5 Conclusion

This review investigated the prior work using the LEGO Mindstorms series and Raspberry Pi systems in regard to the utilization of useful material to be used in the construction of a modular expansion of the Arduino based Plot Clock. Since there is no existing work on the robot, design and construction would need to be conducted to establish a working proof of concept before a proposed modular environment can be incorporated. Furthermore, this document also investigated the open source material and schematics provided by Heberlein to ascertain how the physical hardware component interacted with the software component.

References

- [1] J. Heberlein, "PlotClock by joo - Thingiverse." Online. Available: <http://www.thingiverse.com/thing:248009/>. Accessed: 2015.05.26.
- [2] J. Heberlein, "Ding:Plotclock - Wiki fablab." Online. Available: <http://wiki.fablab-nuernberg.de/w/Ding:Plotclock>. Accessed: 2015.05.27.
- [3] J. Heberlein, "PlotClock source files - GitHub." Online. Available: <https://github.com/9a/plotclock/blob/master/plotclock.ino>. Accessed: 2015.05.26.
- [4] "Arduino - Libraries." Online. Available: <http://www.arduino.cc/en/Reference/Libraries>. Accessed: 2015.05.27.
- [5] "Whats NXT? LEGO Group Unveils LEGO® MINDSTORMS NXT Robotics Toolset at Consumer Electronics Show." Press Release. Retrieved from: <http://www.lego.com/eng/info/default.asp?page=pressdetail&contentid=17278&countrycode=2057&yearcode=&archive=false>, January 2006.
- [6] "Research Robots - Adept MobileRobots." Online. Available: <http://www.mobilerobots.com/ResearchRobots.aspx>. Accessed: 2015.05.27.
- [7] "MIT Handyboard and Super Cricket platform controllers." Online. Available: <http://www.handyboard.com/>. Accessed: 2015.05.27.
- [8] F. Klassner and S. D. Anderson, "Lego MindStorms: Not just for K-12 anymore," *IEEE Robotics & Automation Magazine*, vol. 10, no. 2, pp. 12–18, 2003.

- [9] M. F. McNally, F. Klassner, and C. Continanza, “Exploiting MindStorms NXT: Mapping and Localization Projects for the AI Course.,” in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference*, May 2007.
- [10] T.-C. Wan, *FULLY ARMED NXT: ARM Assembly Language Programming Using the LEGO®MINDSTORMS®NXT*. Universiti Sains Malaysia, 2013.
- [11] “LEGO® Mindstorms Education - NXT information.” Online. Available: <http://www.ratolab.be/downloads/lego/NXT%20information.pdf>. Accessed: 2015.5.27.
- [12] M. Overmars, *Programming Lego Robots using NQC*. Department of Computer Science, Utrecht University, The Netherlands, 1999.
- [13] R. F. Guedes, L. P. Reis, and A. J. Sousa, “Efficient Robotics using the Lego NXT Platform and .Net,” in *Proceedings of the Second Workshop on Intelligent Robotics-IROBOT*, 2007.
- [14] H. J. Levesque and M. Pagnucco, “Legolog: Inexpensive experiments in cognitive robotics,” in *In Proc. of CogRob2000*, 2000.
- [15] K. J. O’Hara and J. S. Kay, “Investigating Open Source Software and Educational Robotics,” in *Consortium for Computing in Colleges*, vol. 18, (USA), 2003.
- [16] “Downloads - Mindstorms LEGO.com.” Online. Available: <http://www.lego.com/en-us/mindstorms/downloads/?domainredir=mindstorms.lego.com>. Accessed: 2015.5.27.
- [17] “Support - Mindstorms LEGO.com.” Online. Available: <http://www.lego.com/en-us/mindstorms/support>. Accessed: 2015.05.28.
- [18] “RobotC - Wikipedia, the free encyclopedia.” Online. Available: <http://en.wikipedia.org/wiki/Robotc>. Accessed: 2015.05.28.
- [19] “ROBOTC.net :: Purchasing RobotC.4x for LEGO Mindstorms.” Online. Available: <http://www.robotc.net/purchase/lego/>. Accessed: 2015.05.28.
- [20] “GNAT GPL for Lego Mindstorms.” Online. Available: <http://libre.adacore.com/tools/mindstorms/>. Accessed: 2015.05.28.
- [21] J. Hansen, “NBC Programmers Guide.” Online. Available: http://bricxcc.sourceforge.net/nbc/doc/NBC_Guide.pdf. Accessed: 2015.05.28.

- [22] “NBC - Sourceforge.” Online. Available: <http://bricxcc.sourceforge.net/nbc/>. Accessed: 2015.05.28.
- [23] J. Hansen, “NXC Programmers Guide.” Online. Available: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf. Accessed: 2015.05.28.
- [24] “leJOS - Wikipedia, the free encyclopedia.” Online. Available: <http://en.wikipedia.org/wiki/LeJOS>. Accessed: 2015.05.28.
- [25] D. Blank, L. Meeden, and D. Kumar, “Python Robotics: An Environment for Exploring Robotics Beyond LEGOs,” in *In Proceedings of the SIGCSE Conference*, pp. 317–321, ACM Press, 2003.
- [26] S. Watkiss, “Design and build a Raspberry Pi robot.” Online. Available: <http://www.penguintutor.com/electronics/robot/rubyrobot-detailedguide.pdf>. Accessed: 2015.05.28.
- [27] “Raspberry Pi Comparison Chart.” Online. Available: <http://www.makershed.com/pages/raspberry-pi-comparison-chart>. Accessed: 2015.05.28.
- [28] “GPIO Electrical Specifications - The Raspberry Pi input and output pin voltage and current capability..” Online. Available: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>. Accessed: 2015.05.28.
- [29] “GPIO: Raspberry Pi Models A and B.” Online. Available: <https://www.raspberrypi.org/documentation/usage/gpio/>. Accessed: 2015.05.28.
- [30] “Raspberry Pi - Wikipedia, the free encyclopedia.” Online. Available: en.wikipedia.org/wiki/Raspberry_Pi. Accessed: 2015.05.28.
- [31] “Windows Dev Center.” Online. Available: <https://dev.windows.com/en-us/featured/raspberrypi2support>. Accessed: 2015.05.28.
- [32] “Raspbian Debian optimized for the Raspberry Pi hardware.” Online. Available: <http://www.raspbian.org/>. Accessed: 2015.05.28.
- [33] “Raspberry Pi - Boise State University.” Online. Available: <http://coen.boisestate.edu/ece/raspberry-pi/>. Accessed: 2015.05.29.

- [34] J. Kiepert, “RPiCLUSTER: Creating a Raspberry Pi-Based Beowulf Cluster.” Online. Available: http://coen.boisestate.edu/ece/files/2013/05/Creating.a.Raspberry.Pi-Based.Beowulf.Cluster_v2.pdf, 2013. Accessed: 2015.05.29.
- [35] M. Milosevic, A. Dzhagaryan, E. Jovanov, and A. Milenković, “An environment for automated power measurements on mobile computing platforms,” in *Proceedings of the 51st ACM Southeast Conference*, p. 19, ACM, 2013.
- [36] D. Chheda, D. Darde, and S. Chitalia, “Smart projectors using remote controlled raspberry pi,” *International Journal of Computer Applications*, vol. 82, no. 16, pp. 6–11, 2013.