0.1 Selection of Hardware

0.1.1 MinSeg M2V3 Two-Wheeled Robot

The MinSeg two-wheeled-robot was selected as the designated hardware platform due to:

- Its standard inclusion of several components which are considered desirable with respect to performing a control study. [*See Section 0.1.1.1*]
- Its existing published academic work. [*Howard and Bushnell [1]*]
  [*This highlighted the device as a suitable hardware platform for control studies.*]
- Its existing driver support [*for the Mathworks software environment.*]. [*Mathworks [2]* and *Hurst [3]*]
- Its use of an Arduino-brand single-board microcontroller.
  [*This highlighted a significant level of support for a principal component.*]
- Its relatively affordable cost. [*~$300*]

Specifically, the MinSeg Model *M2V3* [4] was selected as the designated hardware platform, due to:

- Its standard inclusion of 2 [*equivalent but independent*] motoring axes.

n.axes	**Movement**	
1	*One-dimensional*	*(single, straight line only)*
2	*Two-dimensional*	

The MinSeg M2V3 is depicted from multiple angles in Figures **??** - **??**.

0.1.1.1 Components

As stated in Section 0.1.1, the MinSeg M2V3 two-wheeled robot was selected due to its inclusion of all of the desired components to perform a control study. These components are defined in Table 0.1.

Where beneficial, the components in Table 0.1 are described in greater detail in the sections which follow.

Table 0.1: MinSeg Components

Component (Desirable)	**Part Description**
Programmable microprocessor	1x Arduino single-board microcontroller [Mega 2560]
Dual parallel electric-driven traction motors	2x Lego Mindstorm NXT servo motor [*Includes: 1x DC motor, 1x gearbox, 1x encoder*]
Tires with a relatively-high coefficient of friction	2x Lego wheel [2x [43.2 x 28] Balloon Small]
Motor drivers	4x half-H-driver [1x SN754410]
Sensors permitting sufficient observability of:	-
The angular velocity of the wheels	2x Encoder [*See motor.*]
The 3-dimensional position of the body	1x 3-axis gyroscope & accelerometer [1x MPU6050]
PC Communication (*during operation*):	-
Wired	Serial: USB [*Default. Included with microcontroller.*]
Wireless	Serial: Bluetooth [*Supported, but sold separately.*]
Power Source:	-
Wired	USB 5 [V]
Wireless	Battery holster (6x AA) 9 [V]

Component (Unnecessary)	**Part Description**
-	1x Magnetometer [1x HMC5883L]
-	1x Potentiometer [1x ??]

0.1.1.2 Arduino Single-Board Microcontroller

The MinSeg M2V3 is primarily built upon an Arduino Mega 2560 single-board microcontroller. Arduino is company, project, and user-community which focuses on the development of open-source computer-hardware and software with respect to single-board microcontrollers [5]. A major boon of Arduino products is the relatively high level of support which has manifested with their popularity, including (*but not limited to*) the company itself, academic communities, hobbyist communities, as well as third-party private supporters such as math-software company Mathworks.

A brief comparison between the Arduino Mega 2560 and the more standard Arduino Uno is provided in Table 0.2. Most notably, the Mega 2560 has an increased number of input and output interfaces, a superior clock [6] (*not apparent in the table*), and increased memory versus the Arduino Uno.

Table 0.2: Arduino Board Comparison **REF:preliminaryDecisions:arduino:unoVsMega**

Characteristic	**Uno**	**Mega 2560**	**Unit**
Microcontroller	ATmega328 MCU	ATmega2560	-
Programming Interface	USB [*via* ATMega16U2]	USB [*via* ATMega16U2]	-
UART [*Universal Asynchronous Receiver/Transmitter*]	01	04	-
Clock	ceramic resonator	crystal oscillator	-
Clock Speed	16	16	MHz
Operating Voltage	05	05	V
Number of Digital I/O [*Inputs/Outputs*]	14	54	-
PWM	06	14	-
Analog Inputs	06	16	-
Memory/Storage:			
Permanent (*Flash*)	32	256	kB
Permanent (*EEPROM*)	01	04	kB
Working (*SRAM*)	02	08	kB

Due to the inclusion of a USB port (*which is coupled to one of the UARTs*), board-to-PC interfacing (*in either direction*) is relatively convenient, as no special equipment is necessary (*beyond a PC containing integrated development environment (IDE) software*). This applies to programming the board (*via the USB programming interface*), as well as communicating signals during operation.

Photos of the Arduino microcontroller are depicted in Figures 0.1 - 0.2. Additionally, a pin layout is provided in Figure 0.3.

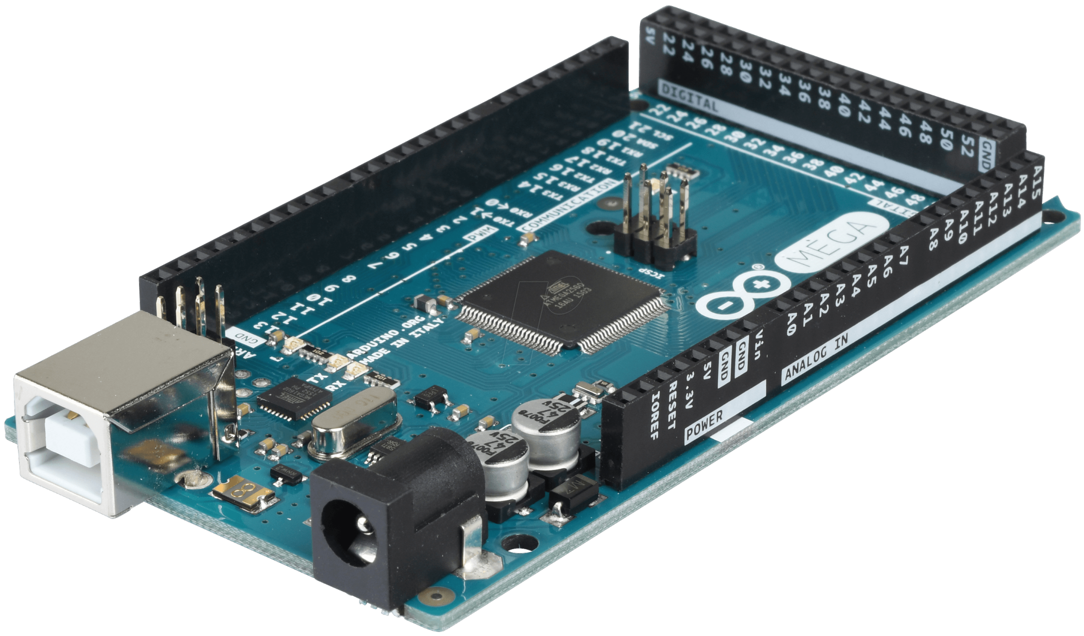

Figure 0.1: [Selection of Compatible HW & SW]: Arduino Mega 2560 (Isometric View) [7]

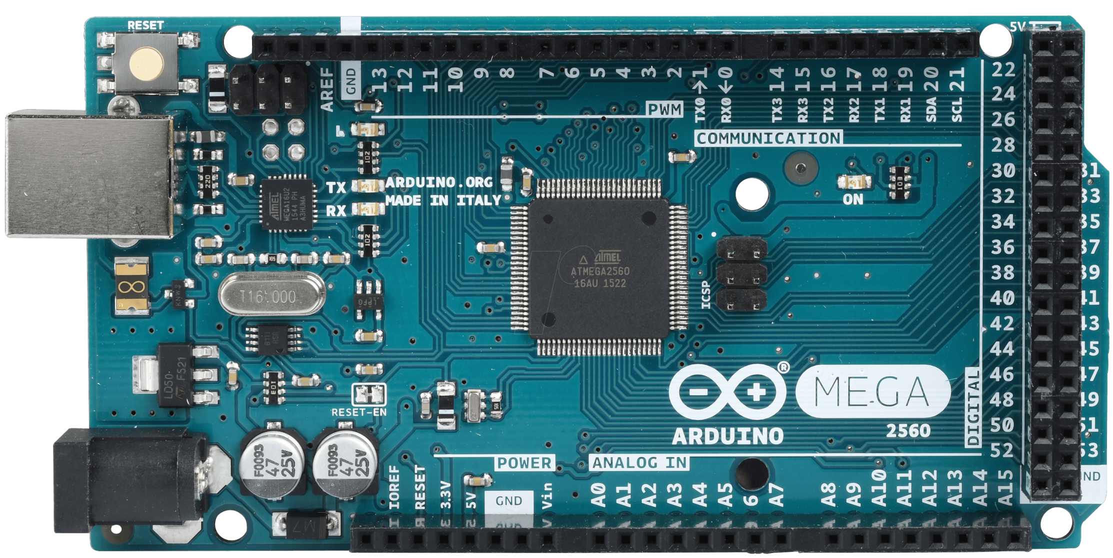

Figure 0.2: [Selection of Compatible HW & SW]: Arduino Mega 2560 (Top View) [7]

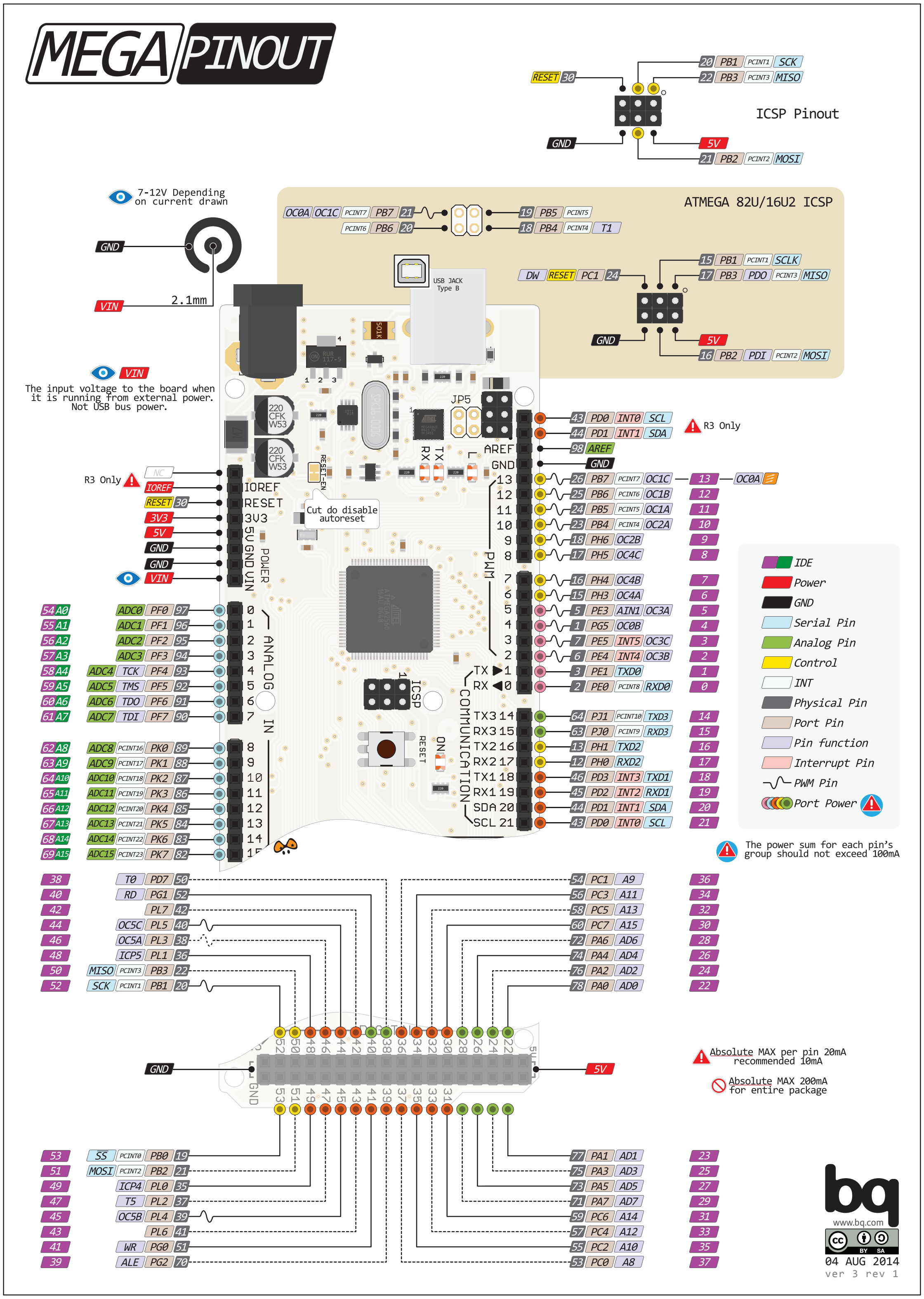

Figure 0.3: [Selection of Compatible HW & SW]: Arduino Mega 2560 (Pin Map) [8]

Digital Outputs and PWM

Serial Controller

-mention input buffer limits

-baud rate limits

0.1.1.3 Bluetooth Module

-

0.1.1.4 Power Source

The MinSeg M2V3 offers two independent sources of power:

- External power via a USB port
- Internal power via an embedded battery holster

A physical switch exists on the MinSeg device to alternate between the two modes of power sourcing.

External-Sourced Power (USB-Cable Connection)

Externally-sourcing power via the USB port offers a constant 5 [V], per the USB standard; however, the cable must be consistently connected to the robot body during use.

Internal-Sourced Power (Battery Pack)

As an alternative to externally-sourced power, power may be sourced from a battery holster embedded within the MinSeg. The battery holster permits the installation of 6 AA-sized batteries.

A typical Alkaline AA-sized battery carries 1.5 [V] at maximum charge. During use, this voltage will rapidly diminish to ~1.25 [V], and more slowly diminish from then on to ~1.00 [V] before rapidly becoming completely discharged, as depicted in Figure 0.4.

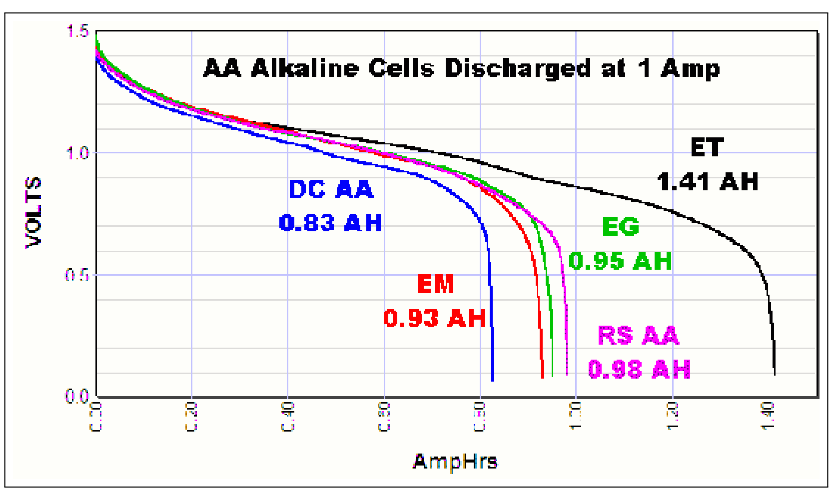

Figure 0.4: [Selection of Compatible HW & SW]: AA-Battery Voltage During Constant Discharge [9]

To use the battery holster as a power source, all six AA batteries must be installed. The batteries are connected in series and therefore cumulatively offer up to 9.00 [V] when at full charge. During typical operation, the batteries will more likely offer a reduced voltage, ~7.50 [V].

Therefore, sourcing power from the battery holster offers consistently greater voltage than external USB-connected sources, (*so long as the batteries are not completely discharged*), and additionally precludes the use of any wiring which could obstruct testing and operation.

0.1.1.5 Motor Driver

The MinSeg M2V3 uses a Texas Instruments (TI) *SN754410*: Quadruple Half-H Driver chip as a motor driver. Supplementary information from the SN754410 datasheet is depicted in Figure 0.5 and Tables 0.3 - 0.6.

Figure 0.5 and Table 0.3 exhibit that the chip has four inputs A and four corresponding outputs Y. Table 0.5 provides a simplified description of the behavior of any one input with respect to its corresponding output:

Input A acts a switch for corresponding output Y:
• If the input pin A is enabled V_{IH}, then the corresponding output Y will output V_{CC2} $[V]$.
• If the input pin A is disabled V_{IL}, then the corresponding output Y will output 0 $[V]$.
Note: It can be assumed that the enable EN is engaged whenever necessary during MinSeg operation.

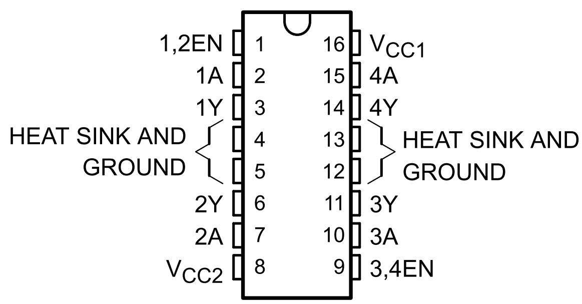

Figure 0.5: [Selection of Compatible HW & SW]: Motor Driver Pin Map [10]

Table 0.3: [Selection of Compatible HW & SW]: Motor Driver Pin Legend [10]

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, non-inverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to circuit board ground plane with multiple solid vias
V_{CC2}	8	—	Power VCC for drivers 4.5V to 36V
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
V_{CC1}	16	—	5V supply for internal logic translation

Table 0.4: [Selection of Compatible HW & SW]: Motor Driver Pin Function Legend [10]

INPUTS(2)		OUTPUTS
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level
L = low-level
X = irrelevant
Z = high-impedance (off)

Table 0.5: [Selection of Compatible HW & SW]: Motor Driver Operating Conditions [10]

		MIN	MAX	UNIT
V_{CC1}	Logic supply voltage	4.5	5.5	V
V_{CC2}	Output supply voltage	4.5	36	V
V_{IH}	High-level input voltage	2	5.5	V
V_{IL}	Low-level input voltage	–0.3(1)	0.8	V
T_J	Operating virtual junction temperature	–40	125	°C
T_A	Operating free-air temperature	–40	85	°C

Table 0.6: [Selection of Compatible HW & SW]: Motor Driver Switching Characteristics [10]

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time, high-to-low-level output from A input	See Figure 3		400		ns
t_{d2}	Delay time, low-to-high-level output from A input			800		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns
t_{en1}	Enable time to the high level	See Figure 4		700		ns
t_{en2}	Enable time to the low level			400		ns
t_{dis1}	Disable time from the high level			900		ns
t_{dis2}	Disable time from the low level			600		ns

Table 0.5 specifies voltages associated with normal chip operation. The voltage source for SN754410 outputs V_{CC2} is wired to the MinSeg power source, and may therefore vary, from $4.5 - 9.0$ $[V]$, (see Section 0.1.1.4).

The SN754410 inputs A are connected to digital output pins on the Arduino microcontroller, specifically those which are capable of producing pulse width modulated (PWM) signals (see Sections 0.1.1.2). Programmed binary lows on the Arduino board will induce 0 $[V]$ and programmed binary highs will induce 5 $[V]$, (*which is the Arduino board operating voltage*).

To achieve voltages other than V_{CC2} exactly, PWM voltage signals are used. The Arduino can set its digital output pin to high for a defined fraction of the time spanning each sample interval of the Arduino board. The effect of the added switching during each sample should be considered minimal, since the MinSeg sample interval operates at the 10^{-3} $[s]$ scale (*as set by the operator*), and the SN754410 switching interval operates at the 10^{-7} $[s]$ scale, (*per Table 0.6*).

As stated in Section 0.1.1.1, there are two DC motors. Each has a positive and negative lead.

0.1.1.6 Motor, Gearbox, and Encoder

The MinSeg implements two Lego NXT servo motors. Each Lego NXT servo motor contains a DC traction motor, a gearbox, and an encoder. A three-dimensional model of the component is depicted in Figure 0.6.

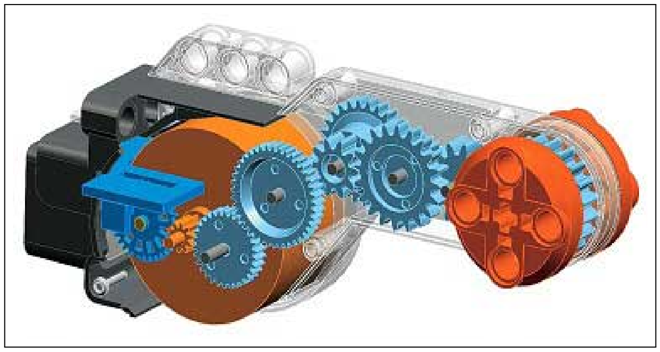

Figure 0.6: [Selection of Compatible HW & SW]: Lego NXT Motor (3D Model) [11]

Although Lego did not publicly disclose all of the characteristic parameters of their components, the hobbyist community reverse engineered several of these values by performing various tests and also by (irreversibly) dismantling a spare [11]. The dismantled component is depicted in Figure 0.7.

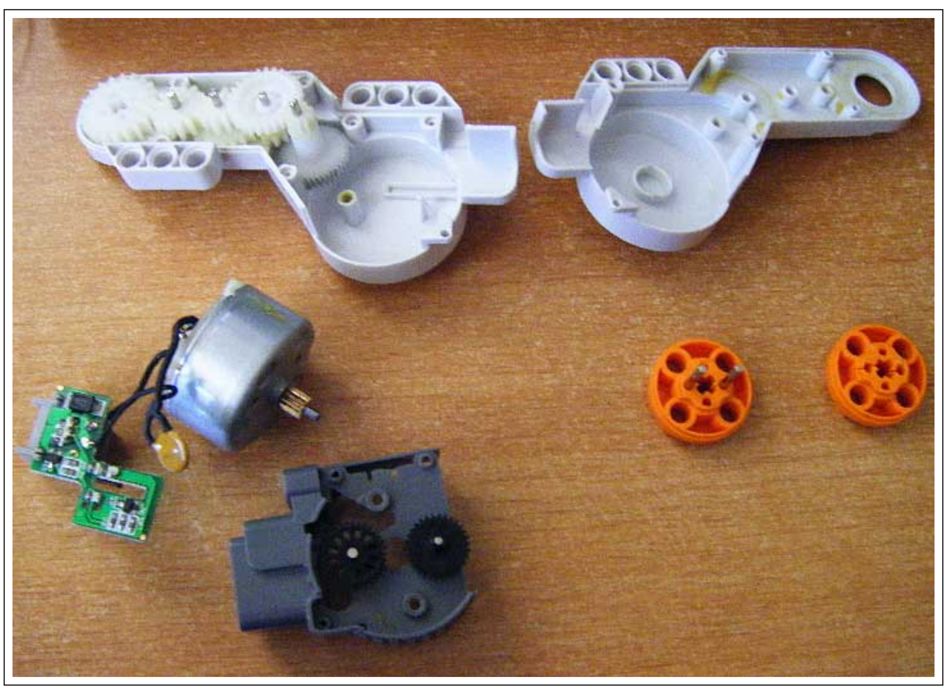

Figure 0.7: [Selection of Compatible HW & SW]: Lego NXT Motor (Exploded) [11]

Table 0.7 provides a legend for the different components in Figures 0.6 - 0.7.

Table 0.7: [Selection of Compatible HW & SW]: Lego NXT Motor Figure Legend [11]

Component	**Figure 0.6**	**Figure 0.7**	
Enclosure	Translucent	Top	Grey
Encoder	-	-	-
Enclosure	Dark blue	Bottom	Black
Gearing	Dark grey	Bottom	Dark grey
PCB	-	Left	Green
Motor	Light orange	Left	Chrome
Gearbox	Light blue	Top-Left	White
Wheel axel mount	Dark orange	Right	Orange

Gearing

The encoder, the motor, and the wheel axel mount are coupled through gearing. Thus, the angular velocity ω of any one of these components can be related to the angular velocity of any one of the other components based on the the number(s) of teeth between each component, as exhibited in Eqn. [0.1], where k represents a ratio and n represents an integer count.

$$k_{\omega\ A2B} = \frac{n_{\omega\ B}}{n_{\omega\ A}} = \left[k_{teeth\ A2B}\right]^{-1} = \left[\frac{n_{teeth\ B}}{n_{teeth\ A}}\right]^{-1} \tag{0.1}$$

The number of teeth in each gearing is depicted in Figure 0.8. Teeth counts in the same row are coupled by teeth. Teeth counts in the same column are coupled by axel, (*and therefore rotate at the same rate, independent of teeth count*).

$$\begin{array}{c|cccccc|c|c}
20 & 13 & 10 & \cdot & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & 20 & 10 & \cdot & \cdot & & \cdot & \cdot \\
\cdot & \cdot & \cdot & 27 & 09 & \cdot & & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 40 & 30 & & 10 & 32
\end{array}$$

Figure 0.8: [Selection of Compatible HW & SW]: Lego NXT Motor Gear Teeth Map [11]

Additionally, in Figure 0.8, each component is separated by a vertical bar. From left to right: wheel axel mount, gearbox, motor, encoder. The completed relation between the wheel axel mount and the gearbox is exhibited in Eqns. [0.2 –0.6].

$$k_{gearTeeth\ wheel2motor} = \left[\frac{13}{20} \cdot \frac{10}{13}\right] \cdot \left[\frac{10}{20}\right] \cdot \left[\frac{09}{27}\right] \cdot \left[\frac{30}{40} \cdot \frac{10}{30}\right] = \frac{01}{48} \tag{0.2}$$

$$k_{gearTeeth\ motor2encoder} = \frac{32}{10} \tag{0.3}$$

$$k_{\omega\ wheel2motor} = \left[k_{gearTeeth\ wheel2motor}\right]^{-1} = \frac{48}{01} \tag{0.4}$$

$$k_{\omega\ motor2encoder} = \left[k_{gearTeeth\ motor2encoder}\right]^{-1} = \frac{10}{32} \tag{0.5}$$

$$k_{\omega\ wheel2encoder} = k_{\omega\ wheel2motor} \cdot k_{\omega\ motor2encoder} = \frac{15}{01} \tag{0.6}$$

Motor

motor torque constant

motor back emf constant

Encoder

slots

binary

quadrature

0.1.1.7 Gyroscope and Accelerometer

precision (250 deg/s)

bias

figure of axes

0.1.1.8 Arduino Interfaces

board sample rate

board run time (inf)

serial to/from board

maximize serial speed

Due to the inclusion of a USB port, interfacing with the board is relatively convenient, whether for programming the board or for data transmission during operation. Additionally, due to the inclusion of multiple UARTs, the board can be configured

-power (see previous section) -communication: serial (usb or bluetooth) -wired or wireless

This can be beneficial when the USB connection is already being used to perform communication; for example, during microcontroller programming or during operation for wired data transmission..

During microcontroller programming, this is especially ideal since the location and, particularly, the attitude of the robot is not significant.

Contrarily, during operation, the robot will need to consistently balance at or near an upright position. While the mass from the cable may be relatively insignificant, tension in the cable could induce significant disturbance forces on the robot during trajectory. These disturbances are beyond the scope of the planned studies, and therefore will be avoided during testing.

0.2 Development PC

Designations pertaining to the development PC with respect to the test platform are exhibited in Table 0.8.

Table 0.8: Development PC Specifications

Hardware	**Version**
PC	2015 Macbook Pro [12]

Software	**Version**
Operating System (OS)	macOS 10.12.5
Xcode *[A Mathworks (macOS)-Supported Compiler [13].]*	7.3.1
Mathworks Software Suite	r2017a
· *MATLAB*	-
· *Simulink*	-
· *Control System Toolbox*	-
· *DSP System Toolbox*	-
· *Instrument Control Toolbox*	-
· *MATLAB Coder*	-
· *Simulink Coder*	-
· *Simulink Desktop Real-Time*	-
· *Matlab Support Package for Arduino Hardware*	17.1.0
· *Simulink Support Package for Arduino Hardware*	17.1.0
Rensselaer Arduino Support Package Library (RASPLib) *[A third-party Simulink Support Package for MinSeg Hardware [3].]*	1.1

0.2.1 Designated PC

A 2015 Macbook Pro PC was selected as the designated development PC, as this was available to the researcher without the need to request additional funding.

0.2.2 Designated Operating System

macOS was selected as the designated operating system, as this was the only operating system installed on the designated PC. (*Version 10.12.5 was the most up to date version at the time of research.*)

Alternative Operating System Compatibility

Although the macOS operating system was used, alternative operating systems (*Windows and/or Linux*) would be equally acceptable.

Such a transition would primarily require an alternative Mathworks-supported compiler [13] which would be compatible with the new operating system. Slight alterations to the method of determining the test platform serial communication channel, as described in Section **??** would also be required.

It is not expected that such a transition would be preventatively difficult.

0.2.3 Designated Hardware-Interfacing Software

The Mathworks Software Suite was selected as the designated hardware-interfacing software due to:

- Its first-party support for programming real-time hardware.
- Its first-party support for simulating real-time hardware.
- Its first-party driver support for Arduino-brand microcontrollers.
- Its third-party driver support for the MinSeg.
- Its first-party support for serial communication with hardware in real-time.
- Its relatively user-friendly language and interfaces.
- Its relative commonality among students and academic institutions.
  [*The software environment was already relatively familiar to the author and to the advising professor prior to performing this study.*]
- Its relatively affordable cost. [*With respect to students and academic institutions. ~$150*].

0.2.3.1 Software Support

0.2.3.1.1 Simulink drivers

0.2.3.1.2 RPI (Joshua Hurst) drivers

0.2.4 Software: Mathworks Software Suite

The Mathworks Software Suite was chosen as the designated software platform due to:

- Its first-party support for programming real-time hardware in a relatively user-friendly language.
- Its third-party driver support for the MinSeg.
- Its first-party driver support for Arduino-brand microcontrollers.
- Its relative commonality among students and academic institutions.
  [*The software environment was already relatively familiar to the author and to the advising professor prior to performing this study.*]
- Its relatively affordable cost. [*With respect to students and academic institutions*].