Bridge Bot User Guide

Aaron Sirken

September 14, 2017

+

Contents

1 Intro **3**

2 Hardware **3**
2.1 Body . . . 4
2.2 Legs . . . 5
2.3 Feet . . . 6
2.3.1 Inserting the motor . . . 7
2.4 Wheels . . . 7
2.5 Motors . . . 8
2.6 JST cables . . . 9

3 Electronics **9**
3.1 Notes on Populating the boards . . . 9
3.2 Base board . . . 10
3.3 Motor Board . . . 11
3.3.1 Motor Board Rev 3.0 . . . 11
3.3.2 Motor Board Rev 3.1 . . . 11
3.3.3 Attaching motors . . . 12
3.4 Radio Board . . . 13
3.4.1 Setting up the ZigBee communication . . . 14
3.5 Battery . . . 15
3.6 Camera . . . 15

4 Software **16**
4.1 Burning the Bootloader . . . 16
4.2 Uploading Arduino Sketches . . . 18
4.3 Interfacing with the Radio Board . . . 18
4.3.1 Example Control Sketch . . . 18

5 Future Work **18**
5.1 Increased complexity of wireless commands . . . 18
5.2 Integrating the I2C motor driver . . . 19
5.3 Integrating the camera . . . 19

6 Board Revision History **19**
6.1 Base board . . . 19
6.1.1 Rev 1 . . . 19
6.1.2 Rev 2 . . . 20
6.1.3 Rev 3 . . . 20
6.2 Motor board . . . 20
6.2.1 Rev 1 . . . 20
6.2.2 Rev 2 . . . 20
6.2.3 Rev 3 and 3.1 . . . 20
6.3 Radio board . . . 20
6.3.1 Rev 1 . . . 20
6.3.2 Rev 2 . . . 21

1 Intro

Bridge bot is a magnetic-wheeled robot designed to traverse and inspect steel bridges. Utilizing custom magnetic wheels, the robot is able to securely adhere to the bridge in any orientation. The body platform features flexible joints connecting the wheels to the body, which allow them to bend up to 90 degrees. This flexibility allows the robot to move across 90-degree plane shifts and traverse various obstacles found on bridges.

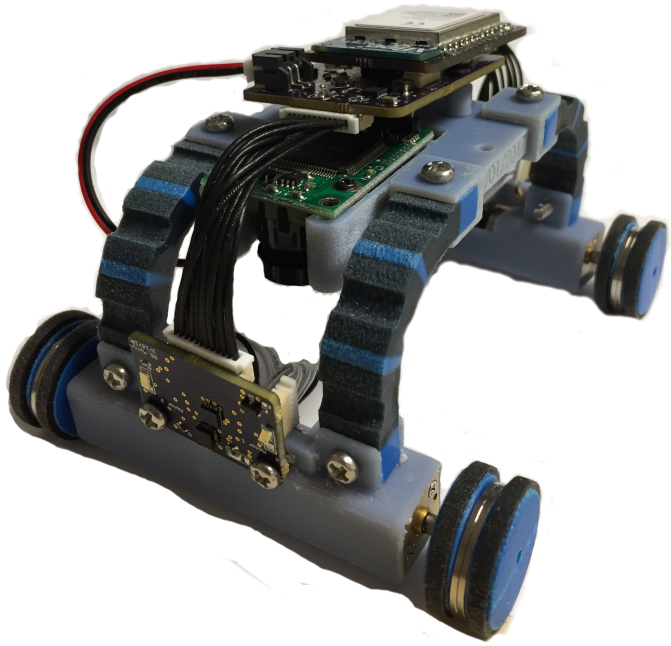

Figure 1: Assembled Bridge Bot

2 Hardware

The Bridgebot is composed of four different 3D printed components. All of the components can be printed on the connex 3D printer and assembled with M3 screws and nuts.

2.1 Body

The body for the Bridgebot can be seen below. It has two mounting holes to attach the base board, a cutout for the camera, battery compartment, and four mounting points for the legs.

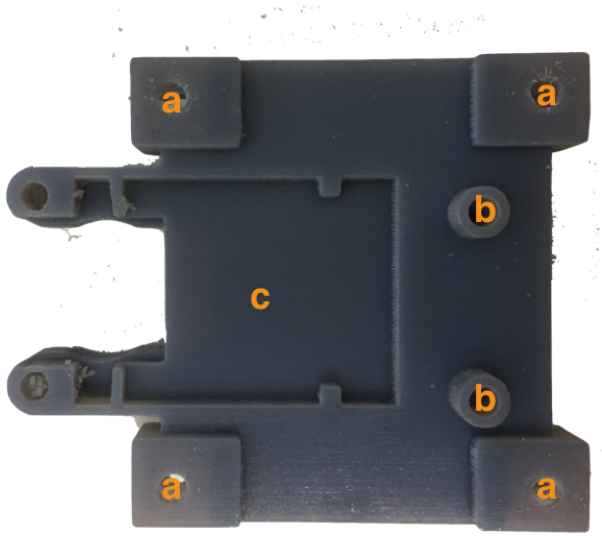

Figure 2: a - leg mounting holes b - base board mounting holes, c - camera mount

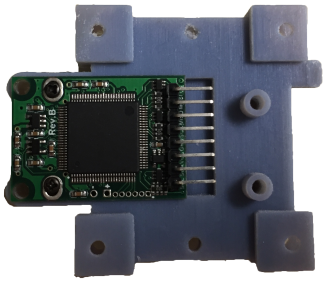

Figure 3: the body with the camera module mounted

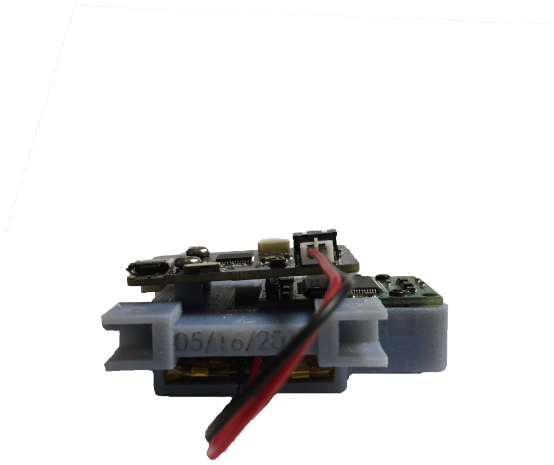

Figure 4: slot for battery

2.2 Legs

The next component of the robot is the leg. The flexible legs are a key feature in allowing the robot to transition planes.

Figure 5: Legs

The leg joints are printed with a gradient of flexible material on the connex. This reduces the stress concentrations in between flexible and rigid materials,

preventing cracking of the material. The specifications for the percent elasticity for the different sections of the leg is specified in figure 6.

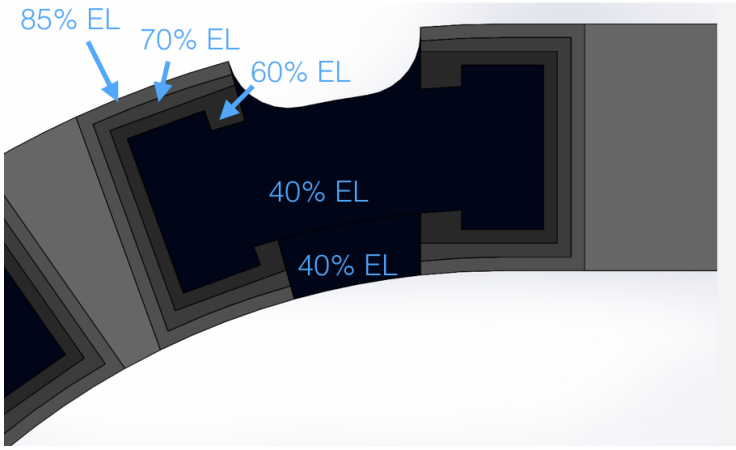

Figure 6: Percent Elasticity specifications for printing the joint

2.3 Feet

The feet of the robot serve as the support structure for the motors. The motors slide into the channels and the motor board is mounted to the feet with M3 screws. Once the motors and motor board have been attached to the, the legs connect the feet to the body of the robot.

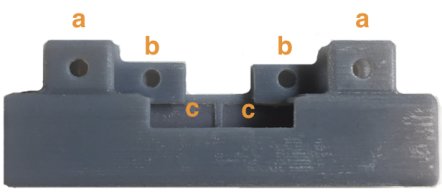

Figure 7: a - leg attachment point, b - motor board mount, c - motor channel

2.3.1 Inserting the motor

To ensure that the motor has a snug fit in the slot, a single layer of double sided tape must be wrapped around the motor before it is slid into the channel.

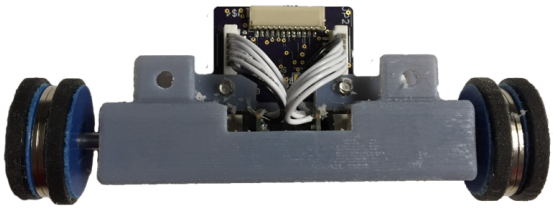

Figure 8: assembled feet with wheels and legs

2.4 Wheels

The wheels are printed in two parts. An axle that attaches to the motor, and a hub that holds the magnets in place.

Figure 9: components of the wheel

Once the wheel has been assembled, the D cutout must be aligned with the motor shaft and press fit into place.

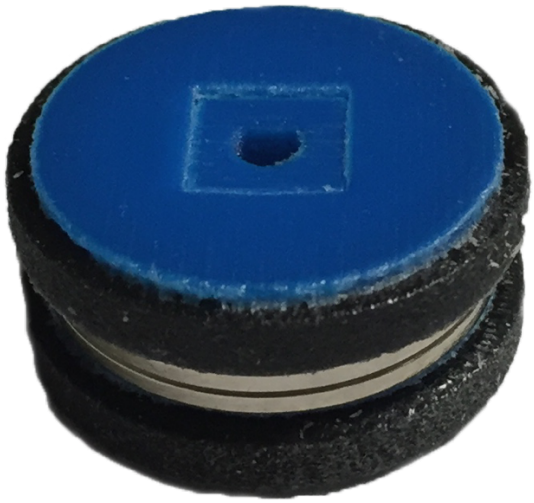

Figure 10: assembled wheel

2.5 Motors

The Bridgebot uses four pololu motors. These motors can be purchased at https://www.pololu.com/product/992

Figure 11: Pololu motor

2.6 JST cables

The JST (japan solderless terminal) cables are used for the motor and base board to communicate with each other. They are custom made by ordering the precrimped wires and female rectangular housing connectors.

It is important to note that the connector orientation has to be reversed in order for the correct pins to match up on the base board and motor board. The figure below demonstrates a properly mated cable. Note that one side has the contacts upwards and the other side does not. 12.

Figure 12: assembled JST cable

3 Electronics

The Bridgebot is composed of three modular circuit boards. Most of the components required to setup the bot have been organized into bags to make assembly easier. They have been separated into: active parts, passive components, connectors, and JST cables. Though these bags contain the majority of the parts required to populate the Bridgebot boards, many of the common resistor and capacitor values are in the grey box with the other SMD components. A complete list of the components required to populate a Bridgebot can be found here:

In addition to all of the electronics, the files for the boards can be found

3.1 Notes on Populating the boards

- Make sure to hand solder the through-holes on the USB as this is the strain relief for the port. Without this the USB will rip off the board when it is removed from the cable

3.2 Base board

The base board of the bridgebot is powered by an Atmel ATSAMD21 processor and is programmed over USB. The base board has 3 JST connections, one for battery, two for motor boards, and one for the camera. Additionally it has a board to board connector used to communicate with the radio board.

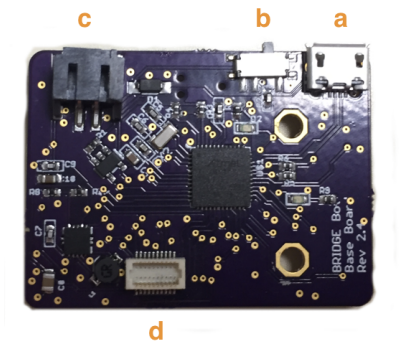

Figure 13: a - usb connection, b - power switch, c - battery JST connection, d - radio board to board connector

Each side of the base board has a 12 pin JST connector that connects the base board to the front and rear motor boards.

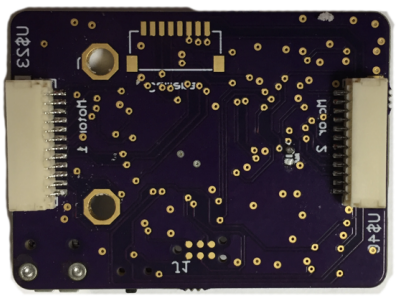

Figure 14: back side of base board with motor board JST connectors

3.3 Motor Board

The motor board contains the motor driver and a 9-axis accelerator, gyroscope, and magnetometer. The motor board converts the signals from the base board to motor commands through a motor driver. It also has LEDs that can be used to illuminate dark areas for the camera.

3.3.1 Motor Board Rev 3.0

This version of the board uses the Alegro3901 driver. The solder jumper on the motor board is to select the least significant bit (1 or 0) on the I2C address of the IMU. This allows both IMUs to be used on the same I2C channel. This motor board does NOT support the data from the encoders.

3.3.2 Motor Board Rev 3.1

The difference between the 3.0 and 3.1 motor board is the motor drivers that are being used. The 3.0 board uses an Alegro3901 motor driver. This driver connects each motor output (8 total) to a pin on the base board. Since the ATSAMD21 chip only has 48 pins, this takes up a lot of the available communications. The solution to this was to use a I2C motor driver (DRV8830). This allowed for more pins to be available to be used for sensing the encoder values. Since the magnetic encoders are quadrature, each motor encoder has two outputs. Since each driver communicates separately over I2C, it is important to be able to select the proper address to use to communicate with them. These boards contain 4 additional solder jumpers that can be used to change the address of each driver. In order to use this driver, a separate library is needed to properly send the I2C signal to activate the correct motor. Currently this library is not developed. This issue is discussed in greater detail in the future work section of the guide.

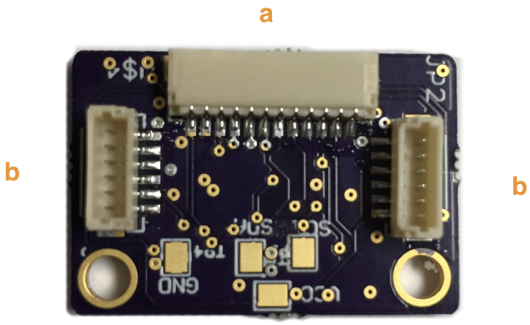

Figure 15: a - base board connector, b- motor connectors

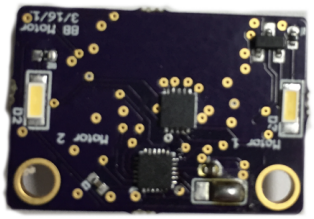

Figure 16: front of motor board - LEDs can be seen on either side

3.3.3 Attaching motors

The motors are connected to magnetic encoders (https://www.pololu.com/product/3081) in order to get better feedback on the motor positioning. To connect the motors to the encoders, simply solder the motor outputs to the corresponding ovular outputs on the encoder PCB.

Following this, the encoder through-hole outputs get connected to a 6 pin JST connector that attaches to the BridgeBot motor board. The JST connection passes the encoders outputs, power, ground, and motor outputs to the main board. Before soldering ensure that the cables are oriented properly to connect to the motor boards. For the Rev 3 and 3.1 boards this means that the black wire (pin 1 on the cable) gets connected to the M1 output on the encoder (the opposite end of the square ground through hole)

Figure 17: motor with jst cable connected - encoder outputs left out for 3.0 rev motor board

NOTE For the current REV 3.0 boards it is important to make sure the encoders outputs are not connected to the motor. The output from the base board was mistakenly connected to both the motor driver input and the encoder output. Leaving the the cable connections disconnected resolves this issue. An updated board is on the repo so future revisions should not have this issue.

3.4 Radio Board

The last Bridgebot circuit board is the radio board. This board attaches to the base board via a board to board connector. To connect it, simply press it onto the base board. The default radio communication for the Bridgebot is 802.15.4 (Zigbee). However,the radio board can be swapped out with different types of wireless communication depending on the application of the Bridgebot.

Figure 18: a - power LED, RX + TX LEDs

The radio board to base board communication has capabilities to speak with any type of wireless device when properly interface. The port has SPI, I2C, 5v, 3.3V, UART, and 2 GPIO pins. This allows it to have a wide range of potential devices to communicate with. Currently, digi offers modules with identical footprints to the currently designed adapter board. These include, Wifi, 2.4GHz, 900Mhz, and an upcoming cellular module. The different xbee modules can be found at:https://www.digi.com/products/xbee-rf-solutions/embedded-rf-modules-modems

3.4.1 Setting up the ZigBee communication

In order to setup the Xbees to communicate with eachother, the Digi XCTU software must be installed. Once this software is installed, this guide

https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu gives an excellent tutorial on how to setup two radios to communicate with eachother.

Since the Bridgebot has its own Xbee module, it cannot simply plug into the USB-Serial converter like other xbee models. In order to connect the radio board to program it, their are exposed pads on the bottom of the board. The figure bellow shows the location of the pads. Simply connect wires to the RX, TX, VCC, and GND and plug them into the corresponding ports on the USB-Serial converter.

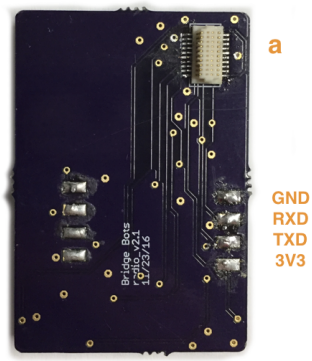

Figure 19: a - board to board connector

This will allow the programming of the Bridge Bot radio boards that is identical to the Digi modules. Once the board has been setup in XCTU, the wires should be removed and the radio board can connect to the base board for wireless communication.

3.5 Battery

The Bridge bot is currently powered by a 500mAH Li-Po battery (ADAFRUIT PRODUCT ID: 1578). The battery connector is a JST PH connector. Though this connector comes with many different batteries, we have found that the Adafruit batteries seem to fit best. When in an ultra low power mode, the ATSAMD21 can reach a sleep state that draws only $70\mu A$ of current. When the chip is active the idle state of the robot draws approximately $10mA$ of current. Thus the estimated idle time of the bridgebot is approximately 24 hours. Future plans will have the robot docked at a charging station so it will never need to reach the maximum duration of its idle time.

The power consumption of the Bridge bot when it is moving along the bridge greatly varies on the envrionmental conditions of the bridge. When moving forward on a level surface, the robot consumes 240 ma. This consumption can go up to around 400 ma when the robot goes through a plane transition. Since the duration of the transition is fairly brief in contrast to the overall travel time of the robot. The estimated battery life of the robot will be just over an hour.

3.6 Camera

Bridgebot uses the Arducam camera mini module which can be found at http://www.arducam.com/tag/arduca mini/.

4 Software

4.1 Burning the Bootloader

Before the base board can be programmed via USB, the chip must be flashed with a boot loader. In order to flash the bootloader to the board, an external programmer and the tag-connect cable must be used. The base board has VCC, GND, SWDIO, and SWCLK exposed on test points as seen below.

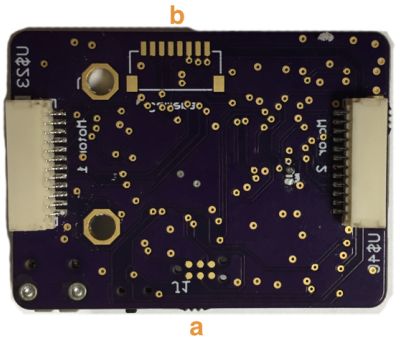

Figure 20: a - programming points, b - camera connection

One side of the tag connect cable attaches to the base board via pogo pins as seen in figure 10. It is held on through a retaining clip pcb.

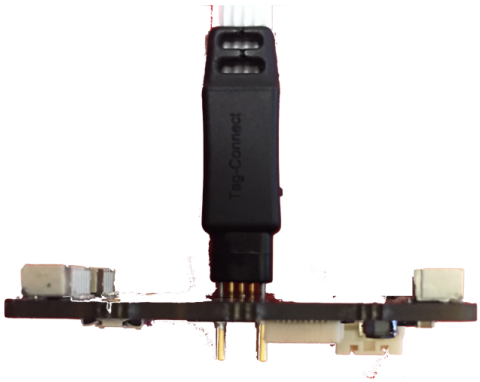

Figure 21: Tag connect attached to base board

Figure 22: Close up of tag connect and retaining clip

The other end of the tag connect cable connects to the J-link programmer.

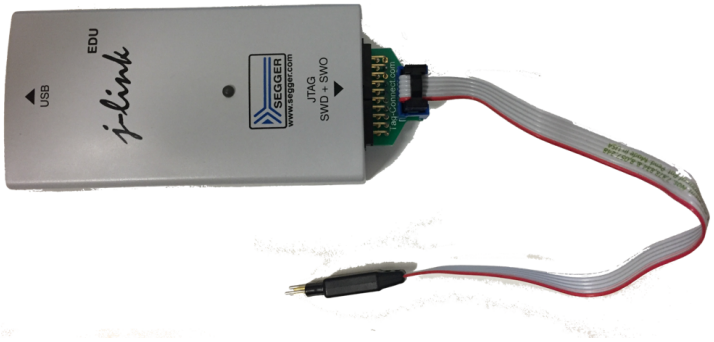

Figure 23: Tag connect cable connected to J-link programmer

Once the board has been connected to a programmer, it needs to be connected to an external power supply (Battery or USB). After it is properly set up, it can be flashed in atmel studio. The hex file to burn to the chip can be found at https://learn.adafruit.com/proper-step-debugging-atsamd21-arduino-zero-m0/restoring-bootloader. Additionally this link contains a detailed tutorial for flashing the bootloader onto the ATSAMD21. This guide is summarized below:

1. Open Atmel Studio
2. Under tools, select Device programming
3. Select the proper programming tool (J-link)
4. Set the device to ATSAMD21G18A
5. Set Interface to SWD
6. Check that the target voltage is 3.3V
7. Go to the Memories tab on the left,
8. Select the .hex file

9. Click program

Now that the board has been properly flashed it can be programmed over USB in the Arduino IDE.

4.2 Uploading Arduino Sketches

After the bootloader has been flashed, the base board will be recognized as an Arduino zero by your computer. To program it, ensure that the Arduino IDE has the Atmel SAMD board files installed. If it has not been installed, go to Tools → Board → Board Manager... → search and install "Arduino SAMD boards". Select the board type in the Arduino IDE as Adruino Zero (Native USB Port) and then upload sketches to the board.

4.3 Interfacing with the Radio Board

The Xbee module is connected to the secondary serial port on the base board. In order to transmit and receive data through the radio, simply use the Serial1.read() or Serial1.write() functions. The normal Serial.write() function will interface with the USB serial port instead of the wireless serial port.

4.3.1 Example Control Sketch

The attached "wireless motor" sketch allows wireless control of the robot. A Zigbee module must be setup with the computer using XCTU and have the complementary settings that ensure that they can transmit and receive data between each other. Simply sending a "H" through the XCTU terminal tells the robot to move forward for a set amount of time. This is the foundation to be able to create more complex commands from a network and can be expanded in future works.

5 Future Work

The current state of Bridgebots has two functional base boards, four functional motor boards, and a single Radio board. This is enough to set up almost two full bots. The second radio board is pending the correct Xbee module.

Since it is sometimes difficult to work with the base boards due to the lack of exposed pins/ports, a great alternative for development is the sparkfun SAMD21 dev board (www.sparkfun.com/products/13672). There are three of these currently in the lab that can be used for development. They have a similar layout to the Arduino Uno but use the same chip as the Bridgebot base boards.

5.1 Increased complexity of wireless commands

The current state of the software allows a command to have the Bridgebot move forward at a set speed. Future development could allow multiple variables passed through the radio that allow the speed and other parameters to be transmitted during commands.

5.2 Integrating the I2C motor driver

The motor board 3.1 revision introduced the DRV8830 motor driver into the Bridgebot platform. Sparkfun developed a minimoto library for this chip

(www.sparkfun.com/products/retired/11890) as they used it in one of their old products. This library works excellent when used on the Arduino UNO, however it does not currently work on the Bridgebot base board. The reason for this is that the Arduino uno is based on an atmel chip and the bridgebot board is based on an ARM chip. The registers and commands in the two chip varients are slightly different and not all libraries are intercompatible.

Even though the sparkfun product is retired as a dev board, seed studio still makes a board based on the same drivers (www.seeedstudio.com/Grove-I2C-Mini-Motor-Driver-p-2508.html). This board has LEDs connected to the motor outputs making it very useful for testing out code.

5.3 Integrating the camera

There has been some development with the Arducam and the Arduino Due boards. Some students working over the summer had sucessfully been able to trasmit data with the zigbee modules. This has not been tested with the new ATSAMD21 chips and has not been fully integrated into the Bridgebot platform.

6 Board Revision History

6.1 Base board

6.1.1 Rev 1

The original design of the based board did not take into consideration that the ATSAMD21 does not come flashed with a bootloader. Without the bootloader, the board was not able to be recognized by the computer. The lack of bootloader was not diagnosed as the initial problem and a full debug of the board was gone through. After diagnosing no problems with the bootloader, the datasheet of the ATSAMD21 was revisited where it was learned that the chips do not come flashed with a USB bootloader. Since the SWDIO and SWCLK pins were not exposed on the chip, it was not possible to burn a bootloader without the use of additional hardware.

Using an external pcb with the correct pins exposed, a bootloader was able to be flashed onto the chip using the instructions detailed in this guide. After flashing the bootloader, the board was recognized by the computer and sketches were able to be uploaded to the board via the Arduino IDE.

The next issue that arose with the base board occured when powering it over battery. Sketches on the board would not run, however the board was still receiving power. After looking at the schematic, the problem was diagnosed as the 5V switching regulator not having a pull down resistor on its enable pin. This caused a floating pin on the enable which put the regulator into an abnormal state, affecting the entire power line of the system.

6.1.2 Rev 2

All of the issues on the previous revision have been rectified. A tag connect pinout has been added to allow for SWD flashing of the bootloader directly to the board. Additionally , a pull-down resistor was added to the enable of the 5V regulator. This solved the power issue and the board now works on battery power. One issue that has been recognized on this board is the polarity of the battery connector. It does not match the default polarity that stock batteries come with and must be fixed on the next revision. In the interim, an adapter that fixes the polarity is being used.

6.1.3 Rev 3

Polarity of the battery connector has been fixed to allow proper connection with off the shelf batteries. JST connectors to motor boards have been combined into a single output. A power switch has been added to the robot.

6.2 Motor board

6.2.1 Rev 1

The initial revision of the motor board successfully controls the motors from the outputs of the base board. One problem that exists with this revision is the I2C communication on the IMU. The IMU can be configured to communicate via SPI or I2c - the Bridgebot uses I2C because of fewer wires. In order to enable the I2C communication, the SPI chip select pin must be pulled high. On this revision it was not. A jumper wire was used to pull the pin high, and successful data transfer occurred.

6.2.2 Rev 2

The chip select pin has been pulled high on this revision. All functionality of the board exists, and there is no need for future revisions in its current state.

6.2.3 Rev 3 and 3.1

The double jst connectors have been merged to a single 12 pin connector. Motors now connect via JST instead of directly soldering into the board. LED have been that can be used to illuminate potential problem areas the camera is photographing. Rev 3.1 switches the motor driver (Alegro3901) to an I2C motor driver (DRV8830). This changes allows for the use of motor encoders with the base board.

6.3 Radio board

6.3.1 Rev 1

The initial revision of the radio board attempted to make a custom board based on the CC2531 MCU. The though behind this was that Resensys uses this chip and it would be the ideal method to communicate with their sensors. Unfortunately, it was not understood that they did not design a custom board around the chip, but rather use a p remade module. This misunderstanding lead to

the attempted development of a custom radio board. Due to the complexity of the board and progamming of the CC2531 chip, the initial board design did not function at all. There was no communication with the chip, and programming the chip directly was poorly documented. Since the radio is a critical part of the robot, an alternative solution of using a premade module was explored.

6.3.2 Rev 2

After the failure of the first design, a simple design on integrating an Xbee surface mount module was explored. This design simply ported the pads on the module to the correct UART, VCC, and GND on the base board. This design allowed for the use of a proven radio module. Additionally, Xbee modules come in Wifi and other radio frequencies and use the same package. This allows the radio module to be swapped out for different communication in different scenarios. After confirming that the Xbee modules would work with the Resensys sensors, we learned about the CC2531 module that they use. It was recommended that the easiest method of integration with their sensors would be to use their module and incorporate the adapter PCB design for the CC2531 instead of the originally intended Xbee.