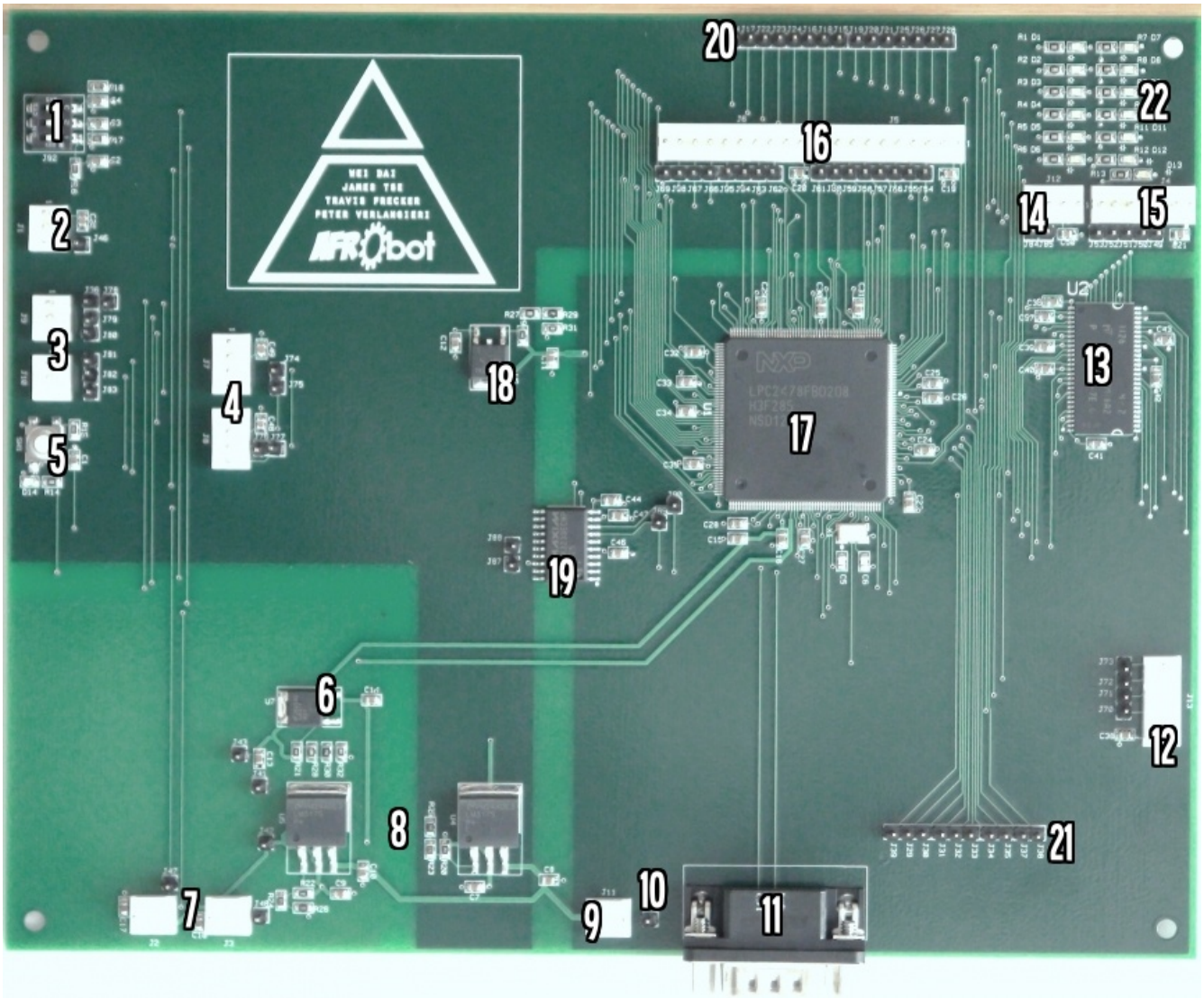
AFRObot Technical Manual

PCB LAYOUT



1. Dip switch	8. 5V voltage regulators	15. Digital compass
2. Servo	9. Battery connector	16. Reflectance arrays
3. Motor controllers	10. Ground pin	17. Processor
4. Encoders	11. DB9 RS232	18. 3.3V voltage regulator
5. Reset switch	12. Sonar	19. Level shifter
6. 3.3V voltage regulator	13. SDRAM	20. GPIOs
7. IR sensors	14. Camera	21. GPIOs (Encoders) 22. Test leds

DESCRIPTIONS

1. Dip switch
Connected to GPIOs for potential use later on (pins 2 and 3). Pin 1 of the dip switch corresponds to pin 110 of the processor, and is the bootloader pin.

2. Servo
Connector for the servo, used for lifting and lowering the writing utensil when tracing lines.

3. Motor controllers
Two connectors for the two motor controllers.

4. Encoders
Two connectors for the two encoders.

5. Reset Switch
The reset for the processor. Active when the green LED next to the switch is on (active low).

6. 3.3V voltage regulator
Connector for the 3.3V analog voltage regulator.

7. IR sensors
Two connectors for the two IR sensors.

8. 5V voltage regulators
Two connectors for the 5V voltage regulators. The left one is analog.

9. Battery connector
Connector for the battery.
NOTE: Only pin 1 works, which is the power pin. The other 2 pins were mistakenly left unconnected, but there is a ground pin (10) immediately to the right of the connector which can be rigged to the connector.

10. Ground pin
Ground access pin.

11. DB9 RS232
Connector for the DB9.
NOTE: The DB9 connector pins are mistakenly reversed. Essentially pin 1 is pin 5, pin 2 is pin 4, pin 3 is pin 3, pin 4 is pin 2, and pin 5 is pin 1 on the top row of the connector. Imagine the top row flipped backwards.

12. Sonar
Connector for the sonar.

13. SDRAM

SDRAM module.
NOTE: Lots of routes are under this module, so it may run hot.

14. Camera
Connector for the camera.

15. Digital compass
Connector for the digital compass.

16. Reflectance arrays
Two connectors for the reflectance arrays.

17. Processor
LPC2478 Processor module.

18. 3.3V voltage regulator
Connector for the 3.3V voltage regulator.

19. Level shifter
The level shifter module.

20. GPIOs
GPIO pins from the processor for future testing and/or unforeseen complications.

21. GPIOs
GPIO pins from the processor for future testing and/or unforeseen complications.

22. Test LEDs
Test LEDs which are connected to corresponding GPIOs for future testing and debugging.

AFRObot Documentation for Source Code

Source Code Overview

3_point
This code is for following the 3 point line on a basketball court, shown in the video demonstration. It's pretty self-explanatory, it follows the same essential line following algorithm.

af
This is the code for the maze following. It basically turns left the first time through in order to solve the maze. So the priority is left, straight, right, and back. Then once the robot arrives at the end of the maze, it will optimize the path by removing redundant paths in the array, such as LBL, replaced with a single S. So on the way back the robot will follow the optimized path.

Elevator
This code was meant to allow the robot to follow a path into an elevator with both the IR line following, as well as using the sonar to detect when the elevator doors would open and close, so that the robot would enter and exit accordingly.

FINAL
This is the code we used for our live demo on presentation day. It essentially follows a line, but ignores any intersections. That made it able to follow a figure-8 for our live demo.

grid
This is the code for grid navigation. Essentially you can hardcode coordinates on an x-y coordinate plane and then it will go to a sequence of coordinates that you specify.

lib
This is the code for the library algorithm. Basically if you reference the video, you can see that this functions closely to the warehouse navigation. It can go through library aisles and mimic the behavior necessary to store and retrieve books.

How do You Run It?

1. Plug in the power/ground for the board at 9 and 10. Power should be pin 1 at connector 9, and ground should be the black pin.

2. Plug in the power and ground for the motor controller on the back of the robot. Ground is black, and power is red. Ground is on the far left, facing the robot from the back, and power is on the far right, from the back.

3. So then you can use the code already on the processor, or you can load your own, or one of the other predefined code packages we've left. To do that, refer to the dip switch at 1, and the push button at 5.

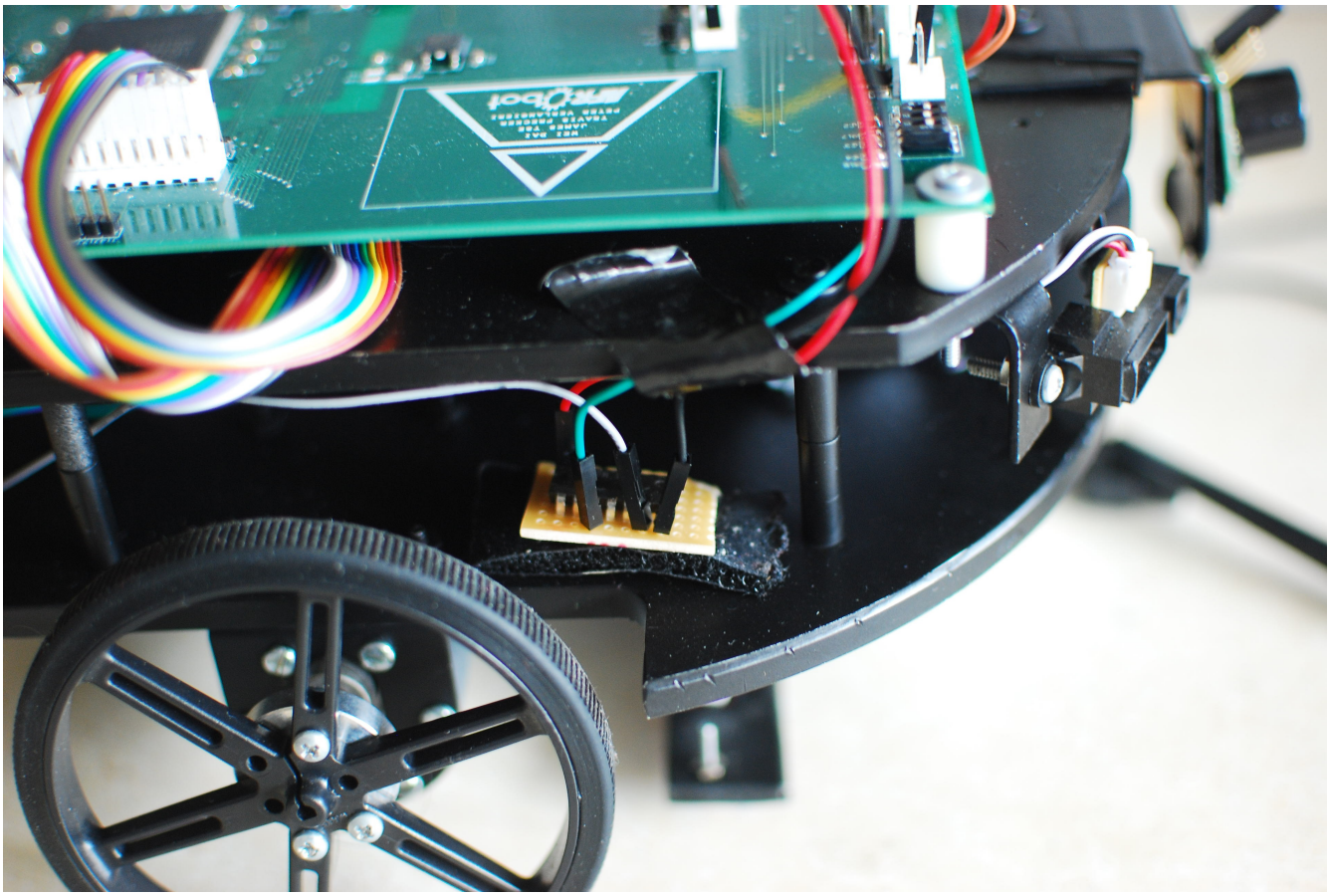
NOTE: The code already on the robot is FINAL. It's the one we used for our live demo.

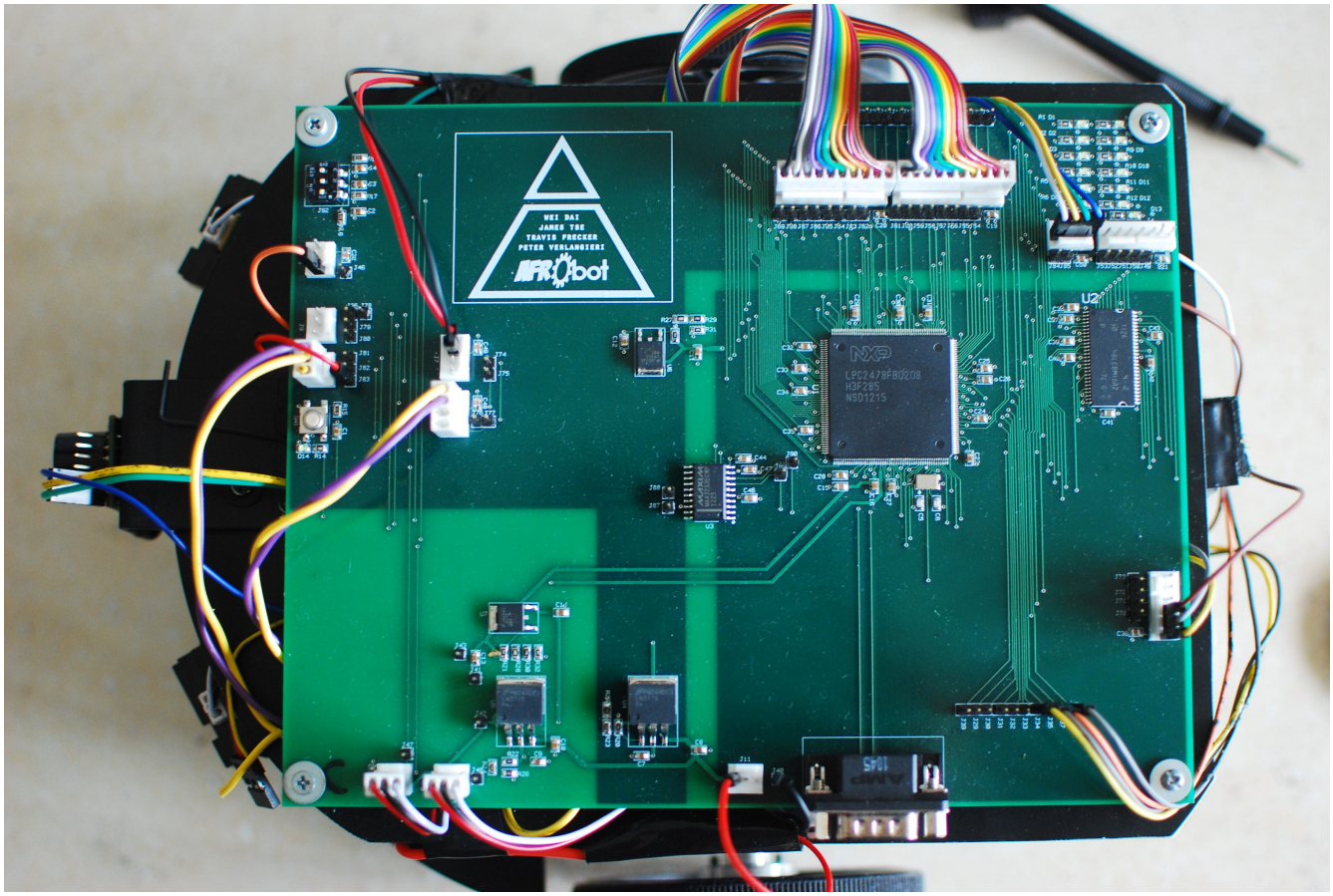
The push button is reset. For the dip switch, all you need is pin 1. When that is set to "on," you're free to flash the board with code via the RS232. When set to off, the robot is free to run.

4. You should be set!

NOTE: We used standard black duct tape for our lines. It's about 1.5 inches thick.

Pictures Showing Connections




WEI DAI
JAMES TOE
TRAVIS FRECKER
PETER VERLANGIERI
MRobot
LPC2478FBD208
H3F285
N5D1215