
ECE 445 Group 22 Fall 2020

**COVID-19 TEST KIT DISTRIBUTION
MACHINE**

Richard Ding, Daniel Polyakov, Brian Zheng

— Introduction —

Current situation:

- Ongoing COVID-19 Pandemic and the possibility of a future pandemic
- Private and public sector need

Idea:

- Automated Machine with authentication that can distribute testing kits
- Can interact with a database on server for verification and testing registration

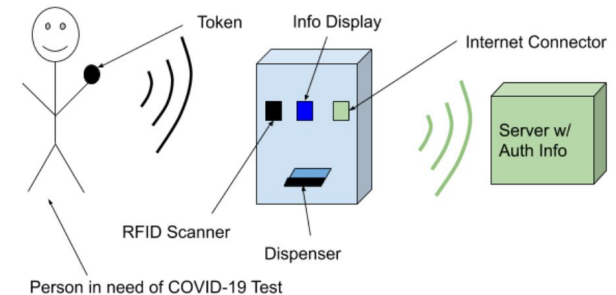


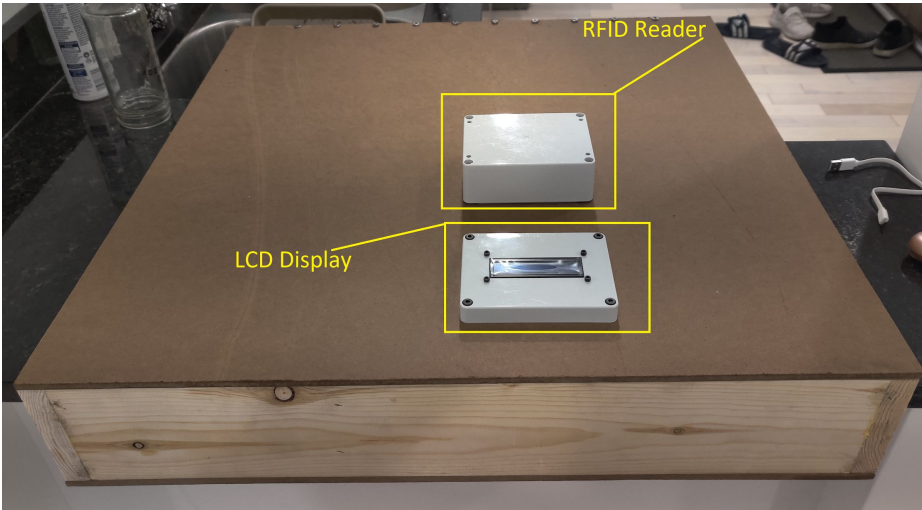
Figure 1: Visual of the Machine and Operation

— Objectives

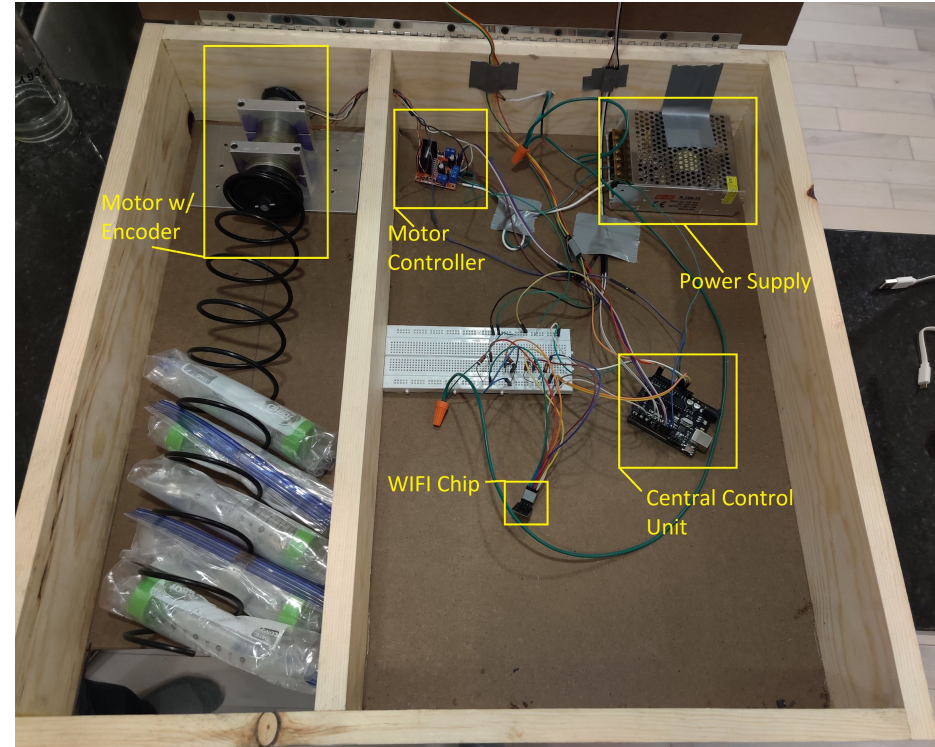
- Labor Reduction -> 0 extra labor assistance
 - Less unsafe contact -> 0 need to touch the machine
 - More security -> 1 Verification token usage for each user
 - More extensibility -> Wirelessly connection enabled
 - More data stability -> Run with SQL, cloud deployable
 - Time saving -> Last no more than 1 min for each user
-

Picture of the Prototype

Outer View

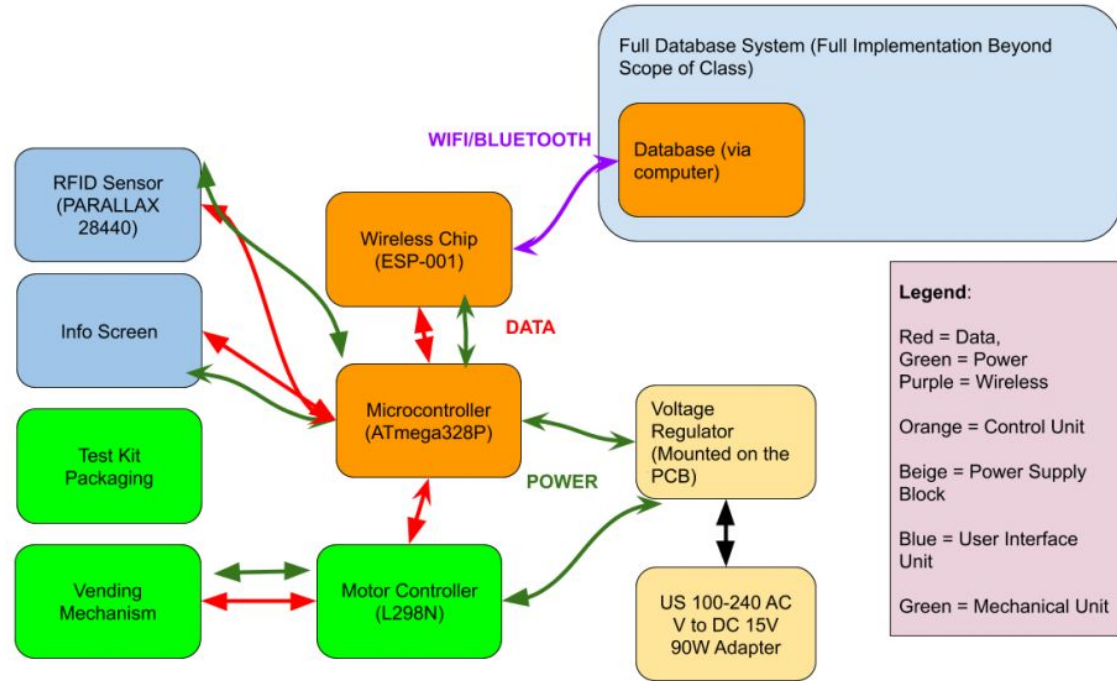


Inner View

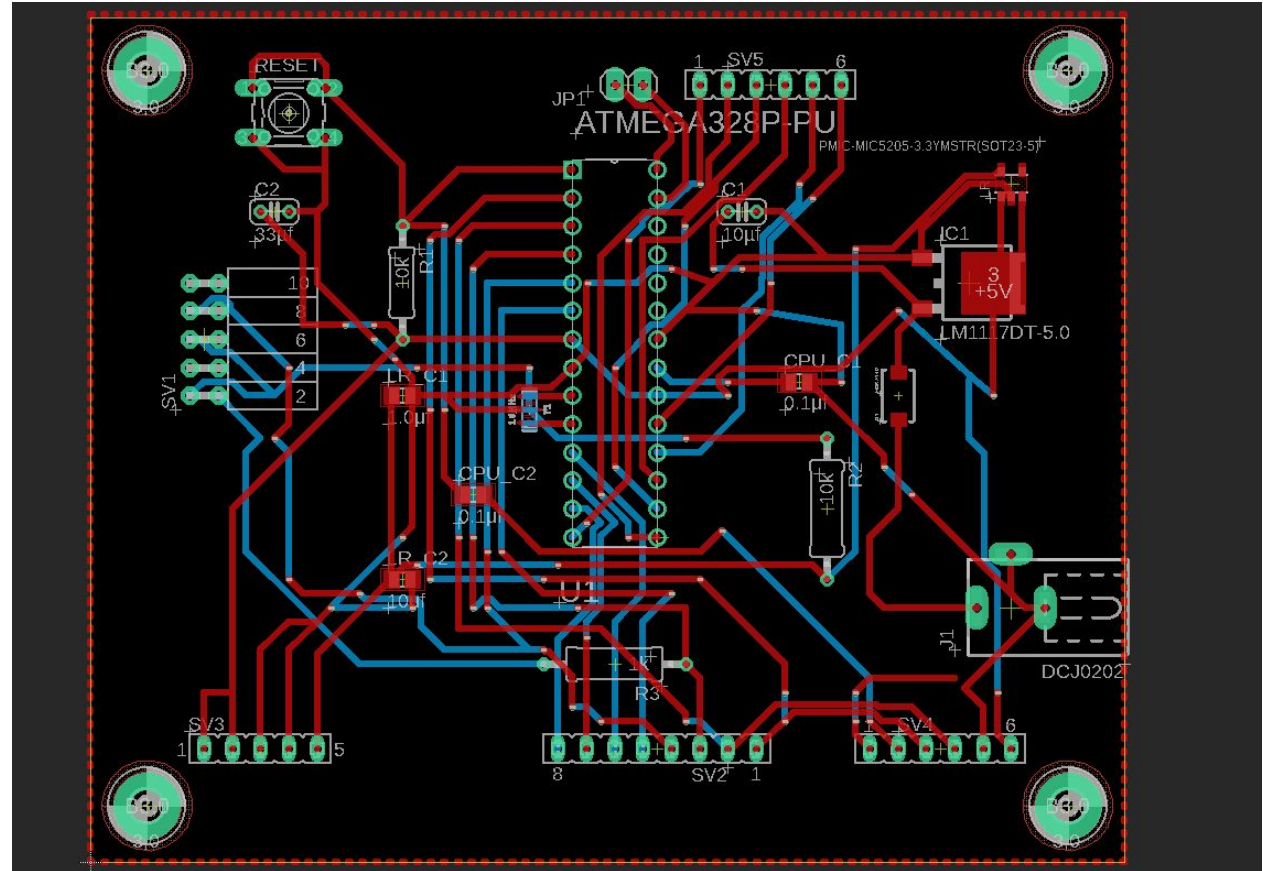


Block Diagram

- Power subsystem
2 Linear Regulators
12 to 5V and 5 to 3.3V
- Control Unit subsystem
ATmega328P 5V
ESP-001 3.3V
- Mechanical subsystem
L298N Motor Controller
A motor with encoder
A spiral
- User Interface subsystem
An info display
RFID Sensor



PCB Design Board View

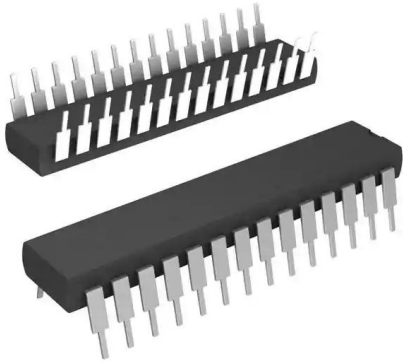




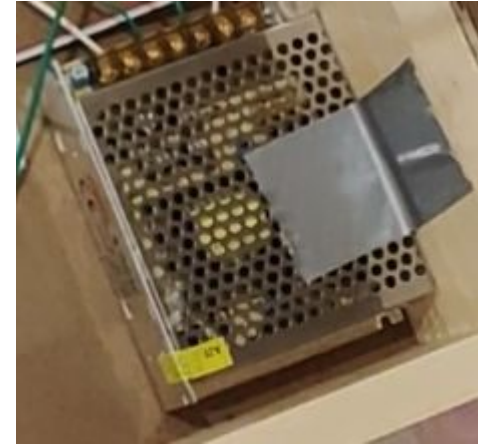
Hardware Overview

(in-house)

- ATMEGA328P-PU
- Voltage Linear Regulators (Convert from 12 V DC -> 5 V DC -> 3 V DC)

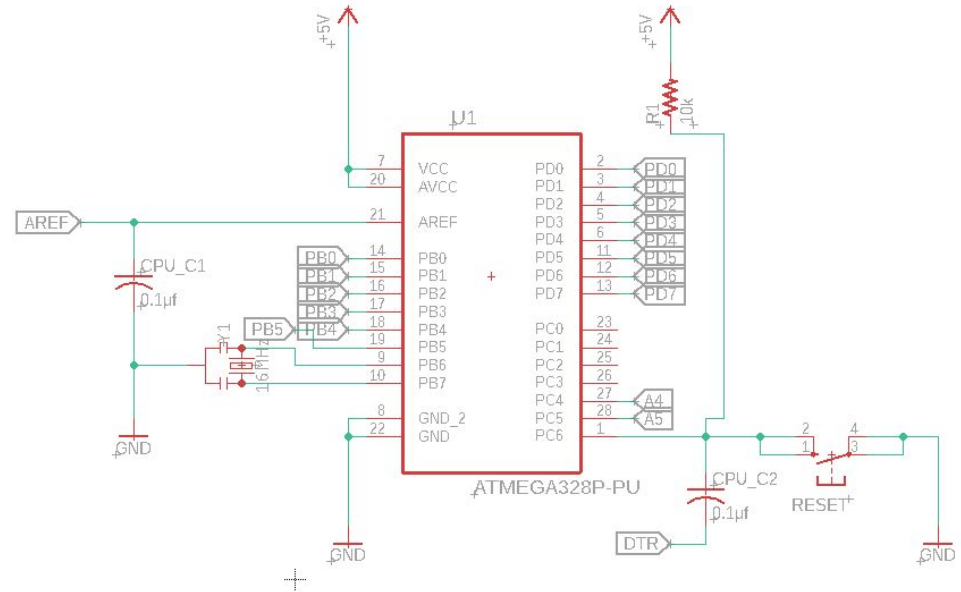


(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ($\overline{SS}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



Central Control Unit (ATMega328P-PU)

- Communicates with all components (RFID, Wi-Fi, and Motor Controller)
- Receive signal from the server, the RFID
- Sends outputs to the L298N motor controller, and communicates with the ESP-001 Wi-Fi Chip

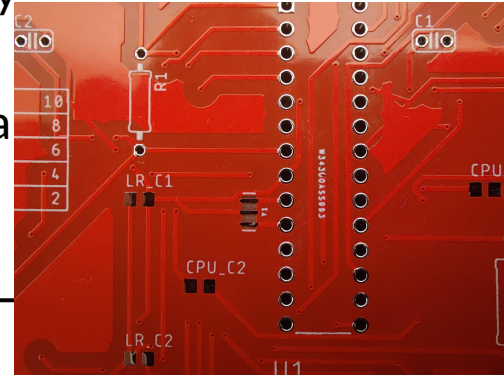
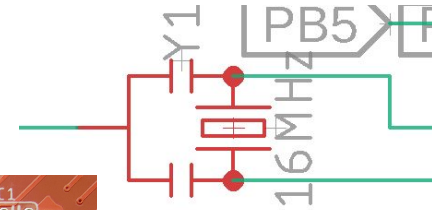
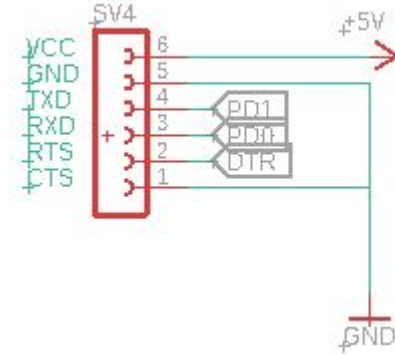


PCB Failure

(ATMega328P-PU)

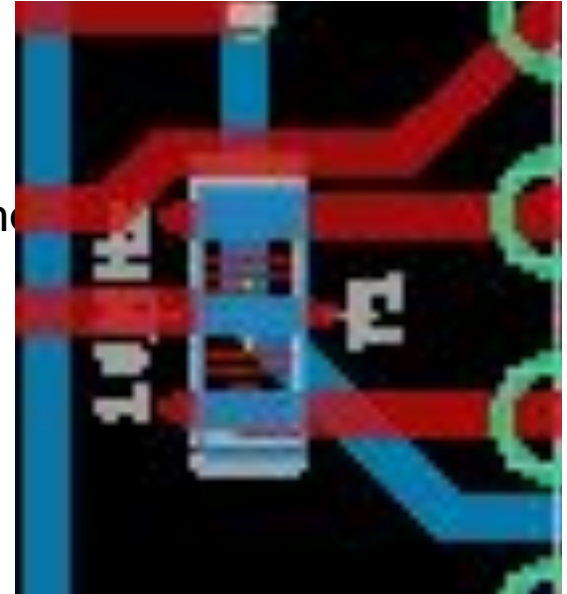
- Programming problems (the USB serial did not interface with board)
- Voltage regulators, capacitors ruled out; all functioned as intended
- Pin connections and ports were correct
- 16 MHz crystal likely shorted due to small size (commonly causes errors if improperly connected)
- USB serial was set to correct port, board, and firmware in the Arduino ISP

Port for the USB Programming FTDI-



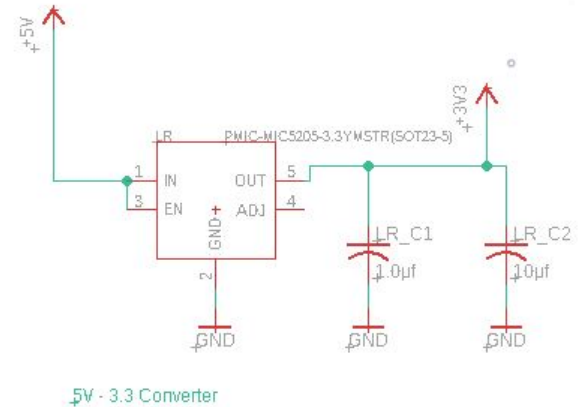
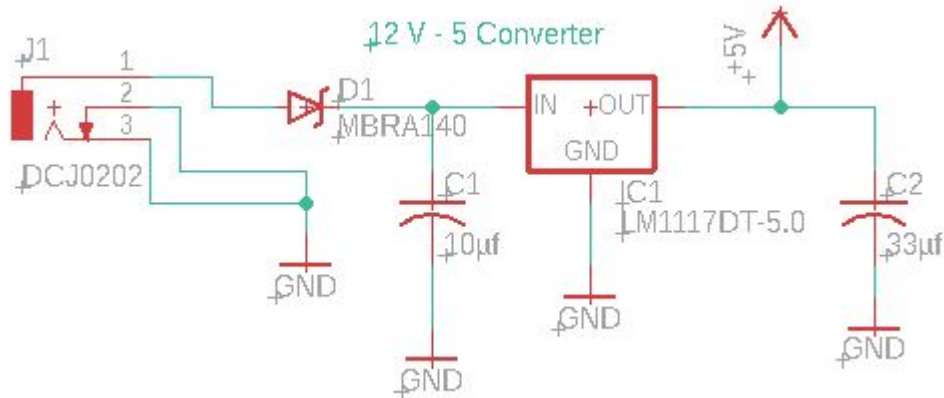
PCB Improvements

- PCB did not make use of a chip mount: easy removal of the microcontroller would reduce time and allow for easier debugging and reprogramming
- Test the USB FTDI beforehand to ensure the ATMEGA328 can communicate
- Use a larger 16 MHZ crystal



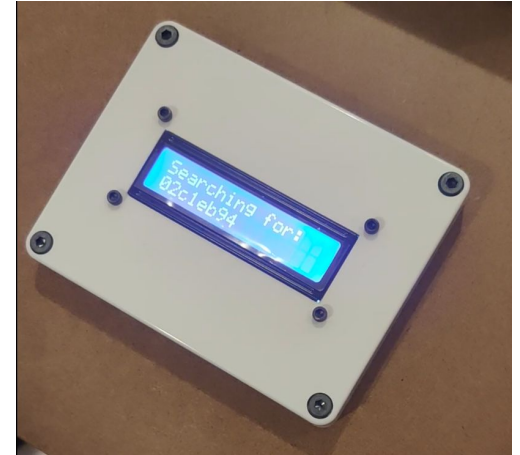
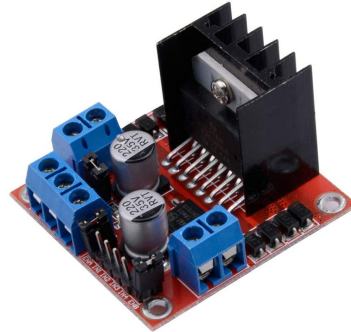
Voltage Linear Regulators

- Convert DC Power 12V (L298N) -> 5V (ATMEGA) -> 3.3V (ESP-001)
- 12 - 5 V (Texas Instruments LM117)
- 5 - 3.3 V (MicroChip MIC5205 Low Noise (3.3 V))



Hardware Overview (Outsourced)

- PITTMAN GM9413D631 Motor with Encoder
- L298N Motor Controller
- Parallax 28440 RFID
- ESP 001 Wi Fi Chip
- LCD
- Power Supply





PITTMAN GM9413D631 Motor with Encoder

- 5 V Encoder
 - 12 V Motor
 - 65:5:1 Ratio
 - Physical calculation of the encoder pulse per revolution (PPR) yielded approximately 1176 PPR (~1200 PPR)
 - Using encoder yields more accurate positioning for the machine to determine when to stop the motor
-

PITTMAN MOTOR Encoder Calculation

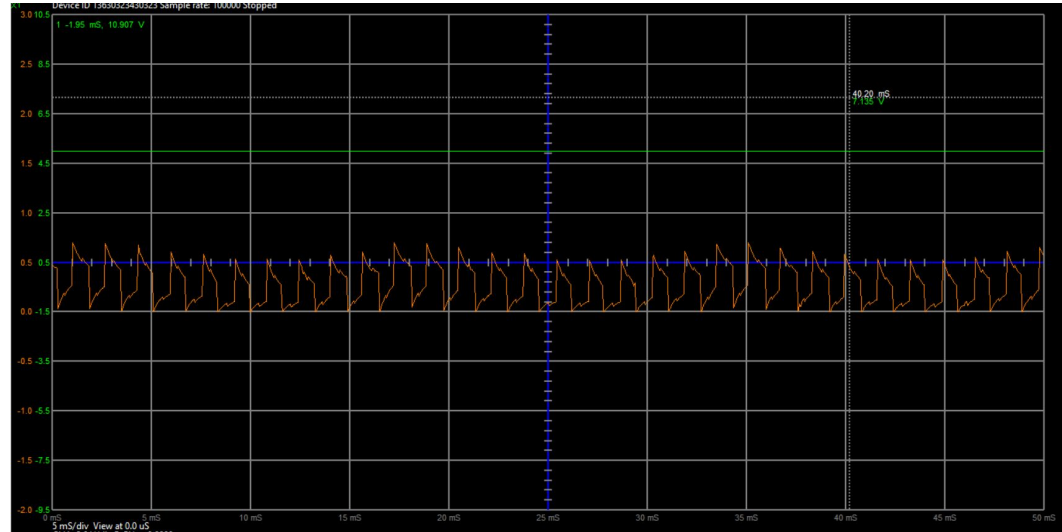
$$\text{Encoder PPR} = \frac{60 \text{ seconds}}{\text{rpm}} \times \text{pulse per second}$$

- To calculate the proper spin of the motor, one must calculate the total amount of pulses needed to conduct a 360 degree rotation (Pulses Per Revolution or PPR).
 - Most encoders come with two outputs, only one output needed (only one direction for the motor)
-

Motor Calculation (continued)

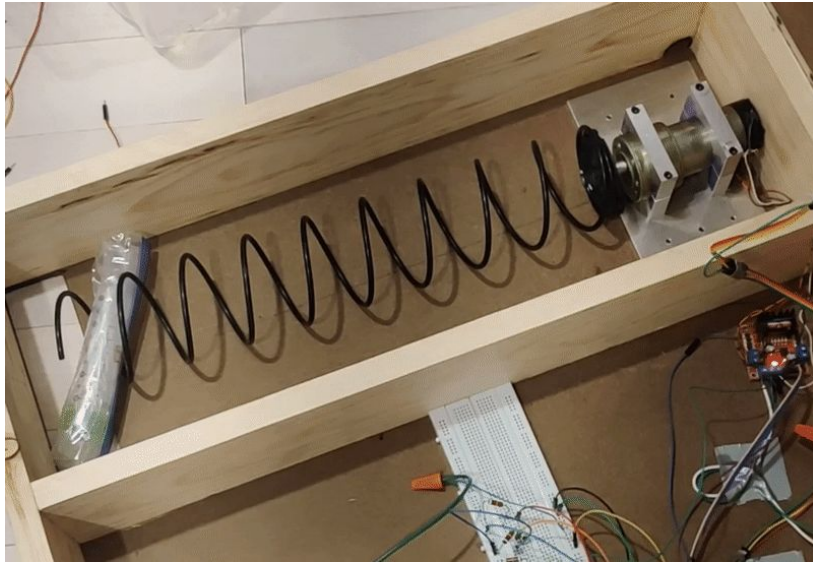
- Using a 5V source to power the motor:
- 30 pulses in 50 mS (or 600 pulses in 1 second)
- RPM: 30.6 rotations per minute.
- Resolution of encoder: 360 degrees/ PPR = .3 degrees per pulse

Measurement of Encoder Pulse (30 pulses over 50 mS)



$$Encoder\ PPR = \frac{60\ seconds}{30.6\ rpm} \times 600\ pps = 1176\ PPR$$

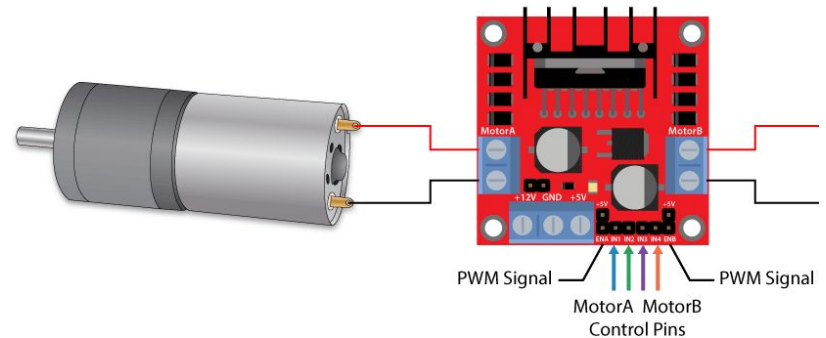
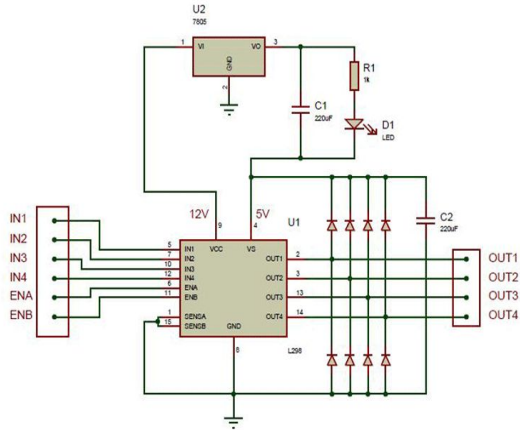
External Factors that can affect a motor's speed



- Friction (what the spiral holds can affect the weight and tension)
 - Temperature
 - Connections (voltage and power supplied to the controller)
 - Winding
-

L298N Motor Controller

- 12 V (more than enough power)
- Provides the power needed to operate the motor
- Can also serve as an auxiliary power for the central control unit if the external power supply fails (via 5V output)



External Power Supply



- LEDMO Switching Converter, AC/DC Power Supply Adapter Transformer Driver for LED Strip Lights, AC 100V/240V to DC 12V, 10A, 120W
 - Serves as the main source of power for the central control unit
 - US Plug
 - Connected directly to the PCB DC power port
-

Parallax 28140 RFID

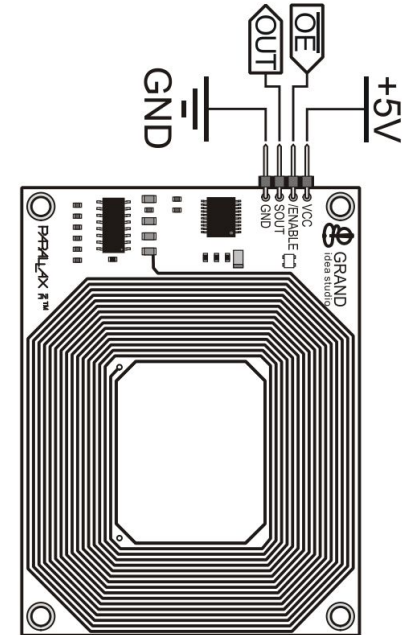
- Necessary for reading RFID tokens from the machine
- 125 kHz
- 5V Input

Parameter	Ideal Value	Tolerant Range	Test Value
V _{CC}	5.0 V	5.0 ± 0.5 V	5.07 V
I _{CC}	108 mA	108 + 92 mA	112.8 mA

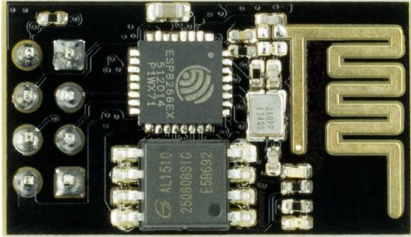
Verification Table

At V_{CC} = +5.0V and T_A = 25°C unless otherwise noted

Parameter	Symbol	Test Conditions	Specification			Unit
			Min.	Typ.	Max.	
Supply Voltage	V _{CC}	---	4.5	5.0	5.5	V
Supply Current, Idle	I _{IDLE}	---	---	9.4	---	mA
Supply Current, Active	I _{CC}	---	---	108	200	mA
Input LOW voltage	V _{IL}	+4.5V ≤ V _{CC} +5.5V	---	---	0.8	V
Input HIGH voltage	V _{IH}	+4.5V ≤ V _{CC} +5.5V	2.0	---	---	V
Output LOW voltage	V _{OL}	V _{CC} = +4.5V	---	---	0.6	V
Output HIGH voltage	V _{OH}	V _{CC} = +4.5V	V _{CC} - 0.7	---	---	V



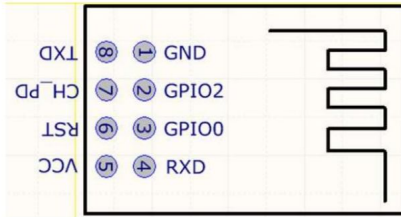
ESP-001 Wi-Fi Chip



- WiFi Protocols 802.11 b/g/n
- Frequency 2.4GHz
- Operating Voltage 3.3V

Operating Condition	Ideal Value	Tolerant Range	Tested Value
Operating Temperature	20 °C	-40 °C to 125 °C	25 °C to 36 °C
Supply Voltage	3.3 V	3.3 ± 0.3 V	3.28 V

Verification Table



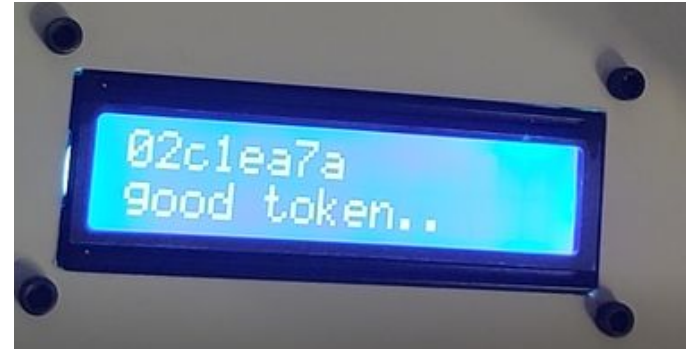
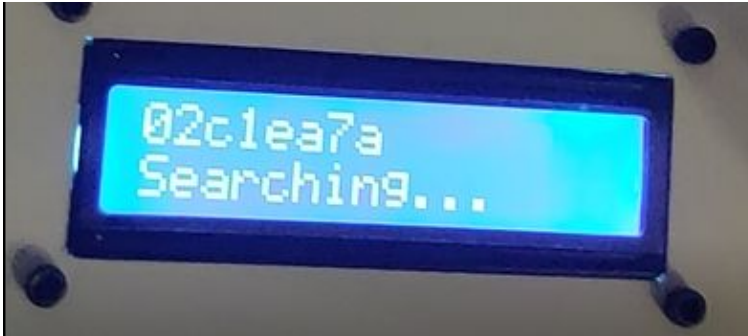
Operating Condition	Symbol	Min	Typ	Max	Unit
Operating Temperature		-40	20	125	°C
Supply voltage	VDD	3.0	3.3	3.6	V

Datasheet Table

LCD

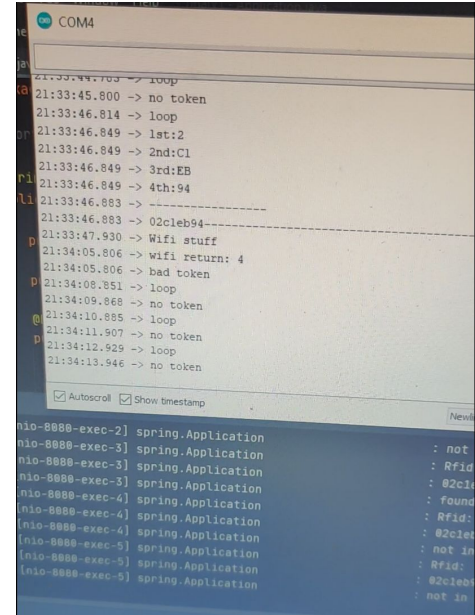


- SunFounder IIC I2C TWI 1602 Serial LCD Module Display
- Standard interface into ATMEGA



Software Overview

- Software in machine
 - ATMEGA running code to manage the RFID, LCD, Motor controller
 - Serial communication with the WIFI chip
 - ESP WIFI running code to manage web communication with server
- Software off machine
 - Authentication Server
 - Running in Java Spring boot



The image shows two overlapping windows. The top window is a serial monitor titled 'COM4' displaying a log of events with timestamps and status messages. The bottom window is a terminal showing the execution of a Java Spring Boot application across multiple instances.

```
COM4
21:33:44.703 -> loop
21:33:45.800 -> no token
21:33:46.814 -> loop
21:33:46.849 -> 1st:2
21:33:46.849 -> 2nd:C1
21:33:46.849 -> 3rd:EB
21:33:46.849 -> 4th:94
21:33:46.883 -> -----
21:33:46.883 -> 02c1eb94-----
21:33:47.930 -> Wifi stuff
21:34:05.806 -> wifi return: 4
21:34:05.806 -> bad token
21:34:08.851 -> loop
21:34:09.868 -> no token
21:34:10.885 -> loop
21:34:11.907 -> no token
21:34:12.929 -> loop
21:34:13.946 -> no token

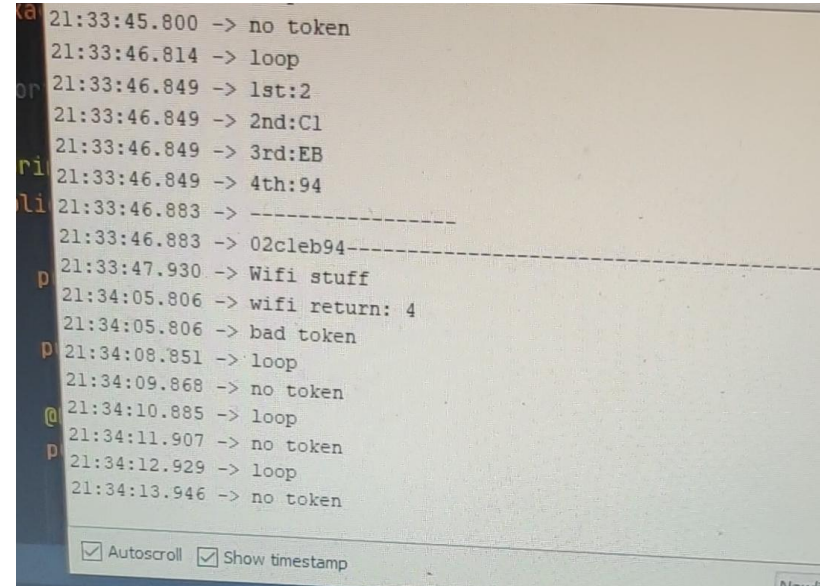
[ ] Autoscroll [ ] Show timestamp

nio-8080-exec-2] spring.Application
nio-8080-exec-3] spring.Application : not
nio-8080-exec-3] spring.Application : Rfid
nio-8080-exec-3] spring.Application : 02c1
nio-8080-exec-4] spring.Application : found
nio-8080-exec-4] spring.Application : Rfid:
nio-8080-exec-4] spring.Application : 02c1e
nio-8080-exec-5] spring.Application : not in
nio-8080-exec-5] spring.Application : Rfid:
nio-8080-exec-5] spring.Application : 02c1eb
nio-8080-exec-5] spring.Application : not in
```

ATmega

Running Code

- Controls RFID and LCD chips
- Interfaces to Wifi Chip on Serial Connection
- Interfaces to L298N Motor Controller
 - Does't send signals to motor directly.
- Directly receives the encoder PPR



```
21:33:45.800 -> no token
21:33:46.814 -> loop
21:33:46.849 -> 1st:2
21:33:46.849 -> 2nd:C1
21:33:46.849 -> 3rd:EB
21:33:46.849 -> 4th:94
21:33:46.883 -> -----
21:33:46.883 -> 02c1eb94-----
21:33:47.930 -> Wifi stuff
21:34:05.806 -> wifi return: 4
21:34:05.806 -> bad token
21:34:08.851 -> loop
21:34:09.868 -> no token
21:34:10.885 -> loop
21:34:11.907 -> no token
21:34:12.929 -> loop
21:34:13.946 -> no token
```

☒ Autoscroll ☒ Show timestamp

ESP-01 WiFi Chip

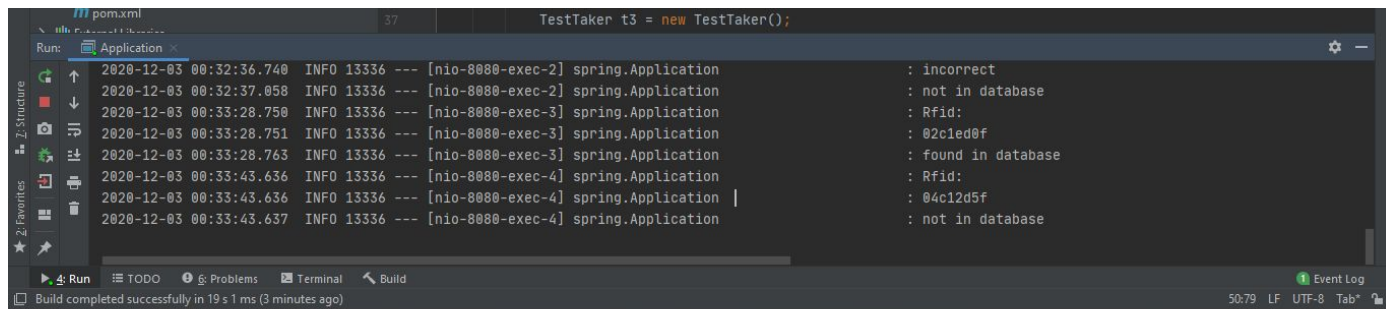
Running Code

- Accepts RFID tokens from ATMEGA, interprets results from server, and returns if valid or not
 - Very common and low cost chip running C code
 - Connects to Wifi then waits for ATMEGA code
 - Hardcoded server host and Wifi ssid/pass
-

Server Side

Running Code

- Accepts RFID tokens, returns if they're valid or not
- REST-ful type endpoint written in Java Springboot
 - Easy to create endpoints and manipulate data
- Running SQL on the backend
 - Highly expandable
- Highly versatile
 - Deployable to your favorite cloud hosting service



The screenshot shows an IDE window with a Java application running. The log output in the console is as follows:

```
2020-12-03 00:32:36.740 INFO 13336 --- [nio-8080-exec-2] spring.Application : incorrect
2020-12-03 00:32:37.058 INFO 13336 --- [nio-8080-exec-2] spring.Application : not in database
2020-12-03 00:33:28.750 INFO 13336 --- [nio-8080-exec-3] spring.Application : Rfid:
2020-12-03 00:33:28.751 INFO 13336 --- [nio-8080-exec-3] spring.Application : @2c1ed0f
2020-12-03 00:33:28.763 INFO 13336 --- [nio-8080-exec-3] spring.Application : found in database
2020-12-03 00:33:43.636 INFO 13336 --- [nio-8080-exec-4] spring.Application : Rfid:
2020-12-03 00:33:43.636 INFO 13336 --- [nio-8080-exec-4] spring.Application : @4c12d5f
2020-12-03 00:33:43.637 INFO 13336 --- [nio-8080-exec-4] spring.Application : not in database
```

The IDE interface includes a sidebar with icons for Explorer, Search, Run and Debug, and Favorites. The bottom status bar indicates "Build completed successfully in 19 s 1 ms (3 minutes ago)" and shows the current file is "pom.xml" with 5079 lines, LF line endings, UTF-8 encoding, and 1 tab open.

—Conclusion—

- Can help government, school and enterprises with COVID testing
- Safe for both human resources and property
- Affordable, about 100-150\$ for basic components
- Easy to assemble and place

Potential Improvements

- Expand the size of the machine to make it contain more testing tubes
 - Construct a more complexed database for a more complete and stable data storage
 - Compatibility and support with tracing apps (i.e. Safer Illinois, COVIDWISE) to record testing information and make testing reminder for users
 - More practical verification tokens to make the system more accessible like ICard
-

Thank You

--GROUP 22
