Memorandum

Date: **April 10, 2023**

To: **Dr. Justin M. Greenly, Ph.D.**

From: **Brendan O'Neill, Jonathan Overbee, Mariano Soares, and Theresa Tine**

Subject: **Introductory Arduino Project Memo**

The purpose of this memo is to outline the intended team meeting times, steps that will be taken if technical or interpersonal problems arise, initial questions from the team, and compiled technical notes of the team on Lesson 2.

Meeting Time: Class times and Tuesdays in St. Junipero Serra Hall at 3:45pm.

Repository

The repository contains a README.md with meeting notes and administrative details such as meeting times. It also contains the lesson documents and will contain the MATLAB code used to drive the robot.

- **Link:** https://github.com/FUSEngineering/EGR16-Spring24-Arduino-Theresa-John-Mariano-Brendan.git

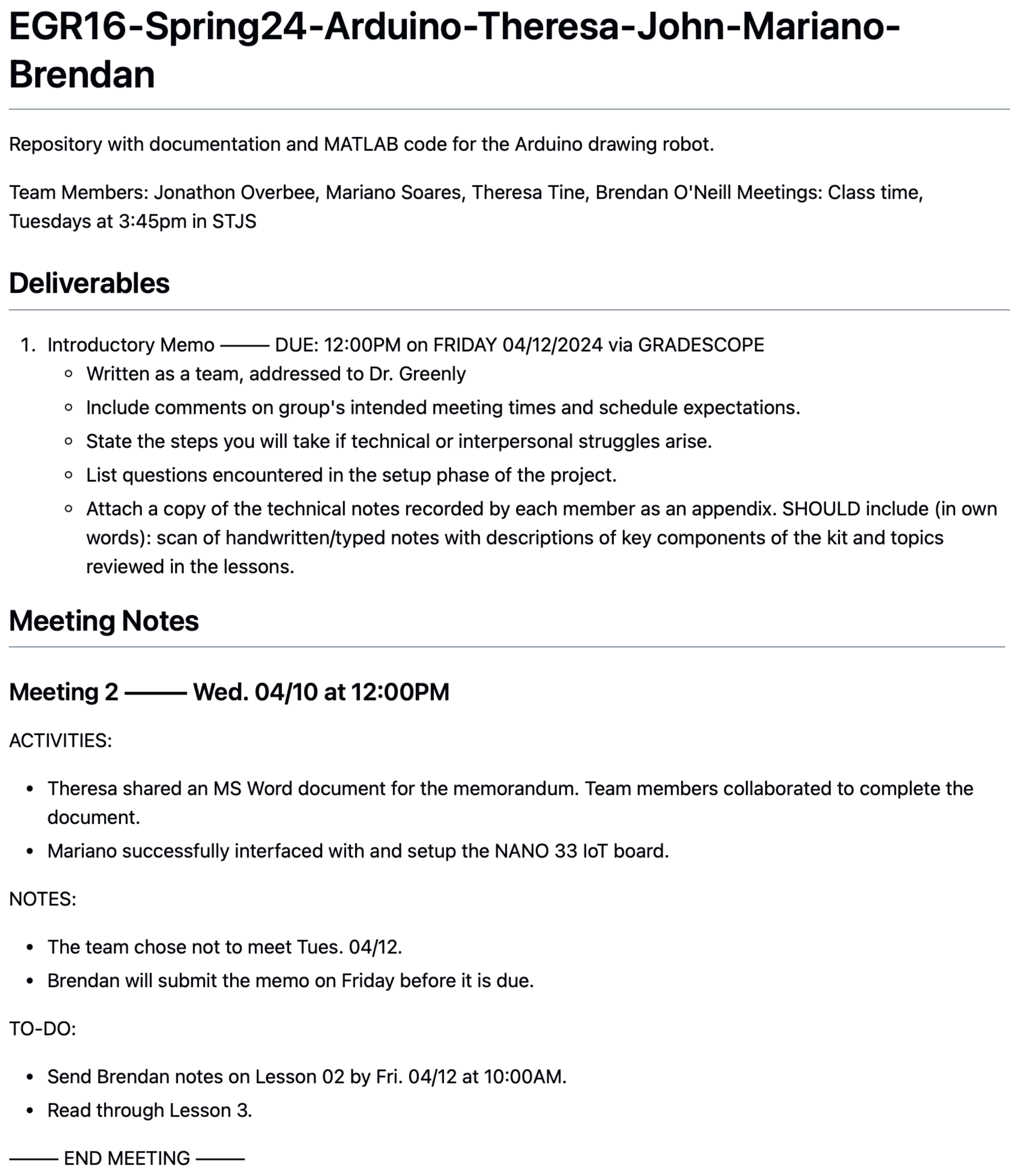
If Problems Arise

- Technical
  - Refer back to the Lesson PDFs, MATLAB official documentation, and Arduino official documentation.
  - Consult other team members, or members of other teams.
  - If all else fails, seek the help of the instructor.
- Interpersonal
  - Discuss the issue(s) as mature adults. If this doesn't work, seek mediation from another team member.

Questions

- On what scale does the robot draw? Only on a small portion of the whiteboard or covering a whole whiteboard?
- What are the recommended hours per week the team should be spending on this project?
- Is there a recommended type of image to draw? What works well with the robot? How much detail is permissible?
- Why do we still use micro-USB?

Team README.md (Screencaptures)



EGR16-Spring24-Arduino-Theresa-John-Mariano-Brendan

Repository with documentation and MATLAB code for the Arduino drawing robot.

Team Members: Jonathon Overbee, Mariano Soares, Theresa Tine, Brendan O'Neill Meetings: Class time, Tuesdays at 3:45pm in STJS

Deliverables

1. Introductory Memo ——— DUE: 12:00PM on FRIDAY 04/12/2024 via GRADESCOPE
    - Written as a team, addressed to Dr. Greenly
    - Include comments on group's intended meeting times and schedule expectations.
    - State the steps you will take if technical or interpersonal struggles arise.
    - List questions encountered in the setup phase of the project.
    - Attach a copy of the technical notes recorded by each member as an appendix. SHOULD include (in own words): scan of handwritten/typed notes with descriptions of key components of the kit and topics reviewed in the lessons.

Meeting Notes

Meeting 2 ——— Wed. 04/10 at 12:00PM

ACTIVITIES:

- Theresa shared an MS Word document for the memorandum. Team members collaborated to complete the document.
- Mariano successfully interfaced with and setup the NANO 33 IoT board.

NOTES:

- The team chose not to meet Tues. 04/12.
- Brendan will submit the memo on Friday before it is due.

TO-DO:

- Send Brendan notes on Lesson 02 by Fri. 04/12 at 10:00AM.
- Read through Lesson 3.

——— END MEETING ———

Meeting 1 ——— Fri. 04/05 at 12:00PM

ACTIVITIES:

- Checking all the components of the Arduino kit and signing the loan agreement.
- Reading through Lesson 01 and installing required MATLAB Add-ons.
- Agreeing upon a meeting schedule (classtime and Tuesdays at 3:45 in STJS) and communication platform (SMS).

NOTES:

- The team decided against using Simulink or the Arduino IDE.

TO-DO:

- Finish configuring MATLAB using steps in Lesson 01.
- Read through Lesson 02 and take detailed notes.

——— END MEETING ———

Lesson 2.1 : The Arduino Envirorment

different boards have different capabilites + functions

↳ different processors, memory, input/output

□ Arduino Nano 33 IoT

↳ IMU Sensor : 3D digital accelorometer + gyroscope

↳ Wifi Module : wifi + bluetooth, local network

↳ Digital Pins : 19 pins, D2 - D21, transmit data + voltage

↳ Analog Pins : 8 pins, A0 - A7, transmit or recive voltage

↳ voltage interpreted as HIGH or LOW w/ threshold btwn 0 and 3.3V

↳ Fixed Function Pins (GND, 5V, 3V + VIN) : ground + 5volt, can provide power to board from externl sourc

↳ Rx/ Tx : communication btwn board + computer

↳ I2C : several devices connected to same line + each device selected by calling its unique address

□ Arduino IDE

↳ open-source Arduino Software (IDE) allows you to write + upload code to the board

↳ Sketch Editor : programs written using IDE

↳ setup() → called once and loop() → called over and over

↳ Tools > Board and Tools > Port menus

↳ Text Console : displays text outputed by IDE

□ Arduino Nano Motor Carrier

↳ add-on board used to control servo, DC, and stepper motors

↳ SAMD11 Microcontroller : recievs comands, control servos, read vales from encoders + battery voltage

↳ servo Outputs : communication btwn servos + Arduino 33 IoT

↳ DC Motor Outputs : connect DC Motors to Carrier Board

↳ DC Motor Drivers : DC Motor control w/ direct connection to Arduino Nano 33 IoT

Lesson 2.1 : The Arduino Enviorment (cont.)

□ Onboard LEDs

which motors are on
↳ Motor LED: indicates which direction spinning

↳ Signal Fault: lights up when fault condition (over current or temp)

↳ On : when board powered

↳ UI : when prob w/ library version incompatable w/ firmware

↳ BAT : when battery being charged

□ Carrier

↳ need to connect Arduino 33 IoT board on the headers @ center of board → ensure proper orientation

↳ when plugged in, some pins unavalible

↳ once board connected, you can start programming the board

□ Charging Battery

1) Arduino Nano 33 IoT connected to Motor carrier
2) Arduino Nano 33 IoT connected to power via USB
3) Power Switch on Motor Carrior is ON

□ Carrier in Drawing Robot

↳ control 2 DC geared motors to control robot position

↳ control servo motor to change marker color

Lesson 2.2 : Getting Started w/ MATLAB

- cd(arduinokit.KitRoot)
- theta = -2*pi : pi/10:2*pi

  ↳ vector theta will represent an angle expressed in radians spanning 2 full revolutions
- Buit-In Functions
  - ↳ sin()
    - ↳ y = sin(theta)
  - ↳ randn( ) : array normally distributded random numbers
- Saving and Loading Data
  - ↳ save myData theta a y z
  - ↳ load myData
- Connecting MATLAB w/ Arduino
  - ↳ "arduinosetup" > Hardware setup > USB > Next > Arduino Board Type (Arduino Nano 33 IoT) > choose port > Select : Motor Carrier, Servo, SPI, I2C > Program > Next

Lesson 2.3 : Getting Started w/ Simulink

- Simulink offers visual programming language (VPL)
  - ↳ based on use of flow programming paradigm to route data, process it, save it, and send outcomes to other programs

□ Blocks and Signals

- ↳ Home > Simulink Start Pg > Blank Model
- ↳ Blocks : used to generate, modify, combine, ouput, and display signals
- ↳ Lines : used to transfer signals from one block to another
- ↳ Signals : flow in the direction indicated by the arrow

Brendan O'Neill - Lesson 02 Notes

Board - Arm Cortex M0+

- wi-fi enabled
- low-cost, high performance 32-bit
- cryptography - secured
- low-power

★ Key Features

↳ IMU Sensor - 3D accelerometer & gyroscope

↳ Wi-fi Module - multi-radio module → Bluetooth & Wi-fi

↳ Pins - transmit & receive voltages for I/O

★ Do not exceed 3.3 volts

↳ RX/TX ports - interface between board & computer

↳ I2C - bus-based serial comm. protocol - SDA & SCL signals

- connect multiple devices to same line

Nano Motor Carrier

- control servo, DC, stepper motors, other actuators

★ Key Features

↳ SAMD11 microcontroller - control servos, I/O

↳ Servo outputs

↳ DC motor outputs

↳ DC motor drivers

↳ ...

Battery Charging

- NANO 33 is connected to carrier & power source
- Carrier power switch is on

In Drawing Robot

- Carrier used to control DC gear motors (x2) and encoder, and servo to change marker color.

MATLAB commands

\`arduinosetup\` opens Hardware Setup menu

...

\`a=arduino\` create arduino object

\`clear a\` releases the object

Notes ARDUINO+MATLAB

2.1 Arduino Envnt

- we use ArduinoNano 33 IoT
  - comes with WiFi chip, crypto chip
  - cheap + used by upcoming engineers

I Arduino IDE

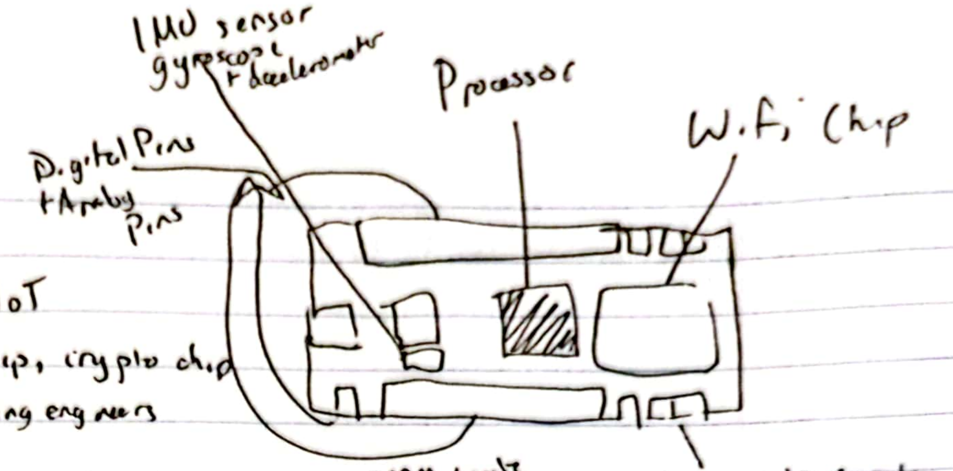
- written in Java

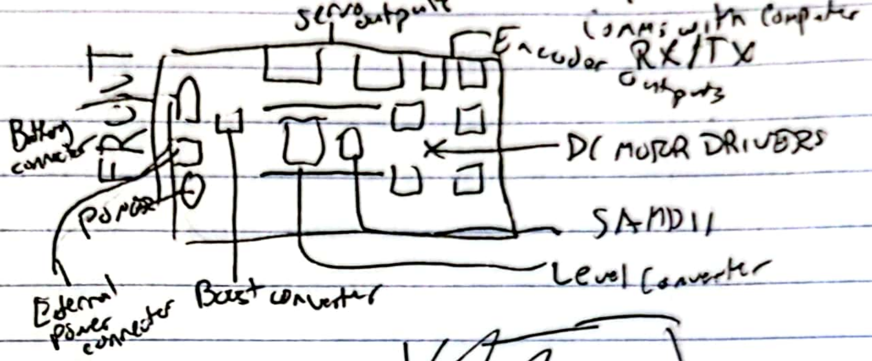
II Arduino Nano Motor Carrier

- controls motors, DC, servo
- goes underneath nano

Parts

- SAMD controlles servos
- Servo outputs handle nano to Servo comms
- boost converter = ↑Voltage
- also has LEDs, analog inputs





2.2 MATLAB

- like Linux for Math
- script or control panel

I. Setup w/ arduino

1. >> arduino setup