MEMORANDUM

To:	Charlie Refvem, Department of Mechanical Engineering, Cal Poly SLO crefvem@calpoly.edu
From:	Antonio Ventimiglia ventimig@calpoly.edu
Date:	September 21, 2025
RE:	**Preliminary Design and Romi Familiarization**

To better understand the Romi, it is best to begin with discussing its main components. The Motor Driver and Power Distribution board is a component used to drive the two DRV8838 motors as well as power other electronics on the robot. Each of those motors have a two-pin interface, one for direction (DIR) and the other for pulse width modulation (PWM). Since each STM32 Timer can control up 4 PWMs, then the wheels can be controlled using one timer.

Let's now focus on the motors, wheels, and encoders provided with the Romi kit. The encoder resolution is 12 counts per shaft revolution. That coupled with the 119.76:1 gear ratio would result in a total of 1437.1 counts per wheel revolution. The smallest increment of distance that the encoder could measure if the Romi is driving in a straight line would then be 0.153mm.

Does having a sensor on each wheel affect the resolution for translational motion? Considering both sensors have the same resolution, there is no gained resolution for translational motion. However, statistically, the average between the two sensor readings would provide a translational motion with smaller uncertainty.

Given that the Romi can be given different types of batteries, it would be best to estimate the maximum translational speed given each type of batteries given. Using a linear extrapolation from the no-load speed of the motor at 4.5V, it can be estimated the no-load speed for six AA NiMH rechargeable batteries (7.2V) and six AA Alkaline batteries (9V) supplied using the below equation:

$$V_{trans,max}\left[\frac{m}{sec}\right] = 150\ [RPM] * \frac{2\pi\ [rads]}{60\ [sec]} * \frac{V_{bat}}{4.5\ [V]} * 0.035[m] \approx 0.12217V_{bat}$$

Using this equation, we get a max speed of 0.8796 m/s for the 7.2V delivered by the rechargeable batteries and 1.0996 m/s from the alkaline batteries, which is 25% faster. This heavily relies on the simplifying assumption that rotational speed scales linearly with the voltage supplied to the motors, which we know not be true.

The Romi does not move holonomically. The Romi may drive any path that lies between its two bounds of motion: circling left (left wheel still and right wheel rolling) and circling right (right wheel still and left wheel rolling). If the wheels were to slip while circling left or right, the bounds of the motion would be different thus changing its shape of path. However, it is best assumed the wheels would only slip in extreme scenarios such as being stuck against a wall since the rubber on the wheels are designed to maintain predictable motion.

Adding an inertial measurement unit (IMU) would provide absolute orientation, angular velocity vector, acceleration vector, magnetic field strength vector, linear acceleration vector, gravity vector, and temperature. This will supplement with the information provided by the wheels encoders since it should provide data unobtainable by the encoders but helps describe the motion of the body.

In terms of line detecting sensors, reflectance sensors (infrared) are best used for following printed/drawn patterns on the ground. They function by sensing the change in reflected light, meaning if the robot moves off path, the black line will no longer be absorbing as much light. A minimum of two sensors are used to track both ends of the black line. They create a "bang-bang" control system that can effectively communicate if the device needs to turn right or left. The sensor placement would have to be pointed along the axis of movement, however it being on the front, bottom, or back shouldn't be important. The MCU would interface with these sensors through input pins. The drawback of too many sensors is that they use up too much of the MCU's bandwidth, preventing you from using them for other functions. Additionally, if the data from the sensor require extensive data processing, then they can take up too much processing power on the device.

I believe, given the nature of the Romi, a microswitch bump sensor would be best. It is simple and robust in nature, providing digital information of collision detection. It would be most useful to mount at the front and front-diagonals of the Romi to detect any head on collisions. Side sensors would be if the device is traverse tight corners. I could be argued that rear sensors can be circumvented by employing movement algorithms that avoid reverse motion.

I don't believe that any other "types" of sensors are required to adequately control the Romi. However, increasing the complexity of the bump sensors to proximity sensors to avoid collision would be ideal, since bumping into every obstacle in the way is unideal.

Table 1. Romi Robot Parameters

Parameters	Value	Units
Chassis Diameter	163	mm
Track Width (Wheel Center to Wheel Center)	141	mm
Wheel Radius	35	mm
Gear Ratio (Exact)	119.7576	-
Encoder Resolution (at Motor)	12	counts/rev
Motor Voltage (Rated)	4.5	V
Motor Torque (Stall)	25	oz-in
Motor Speed (No-Load)	150	RPM
Max Speed (Translational)	0.5498	m/s